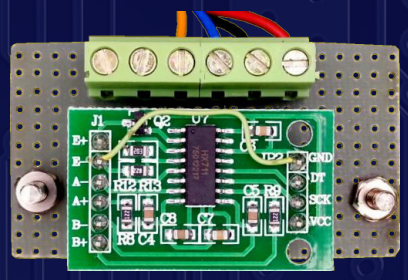
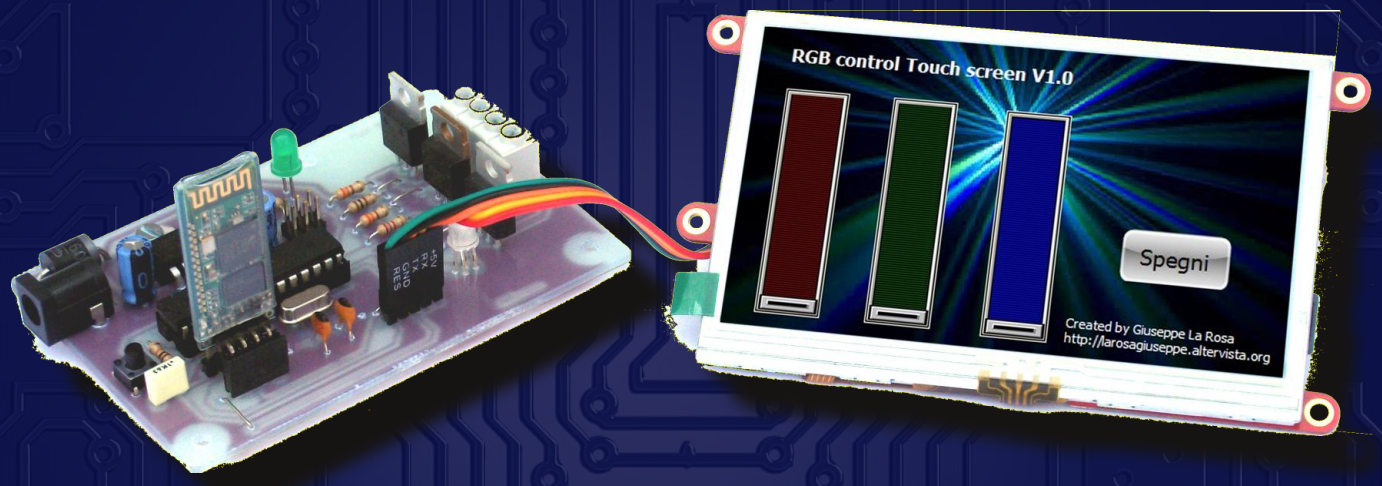
n. 373/374 Dicembre 2016 / Gennaio 2017

Misure di forza con Arduino

ArduLoadCell





CONTROLLO STRIP LED RGB

CON BLUETOOTH O DISPLAY TOUCH SCREEN

- This is (not) Rocket Science (2)
- Il sistema elettrico a 48 Volt
- Registratore di consumi elettrici
- Gestione cisterna per raccolta acqua piovana con Arduino

MOUSER ELECTRONICS

Più prodotti nuovi in magazzino di ogni altro distributore.

Ordinate adesso su mouser.it

SOMMARIO

3 **L'Editoriale**
di Giovanni Di Maria

4 P **MISURE DI FORZA CON ARDUINO: PROGETTO ARDULOADCELL** di Ing. Giovanni Carrera
Scopo del progetto è di misurare forze o deformazioni con un sistema basato su Arduino.

14 P **CONTROLLO STRIP LED RGB CON BLUETOOTH O DISPLAY TOUCH SCREEN** di La Rosa Giuseppe
Questo progetto consente la gestione di una striscia di LED RGB tramite Smartphone Android oppure con un Display Touch Screen.

21 D **THIS IS (NOT) ROCKET SCIENCE - PARTE 2: A PROPOSITO DI BREATHING E PERFORMANCE**
di Lucio Di Jasio
Esaminiamo il mondo di confine tra software e hardware che oggi chiamiamo embedded e che stanno raggiungendo costi sempre più bassi.

26 D **IL SISTEMA ELETTRICO A 48 VOLT: IL PRIMO PASSO VERSO LA MOBILITÀ ELETTRICA**
di Andreas Mangler
Per stare al passo con i limiti sempre più severi imposti sulle emissioni di CO2, il passaggio verso il sistema elettrico a 48 Volt è inevitabile.

31 S **REGISTRATORE DI CONSUMI ELETTRICI**
di Grazia Ancona
Come interpretare e valutare i consumi domestici.

43 S **GESTIONE CISTERNA PER RACCOLTA ACQUA PIOVANA CON ARDUINO** di Girolamo D'Orio
Trattiamo qui della realizzazione di un dispositivo per la gestione di una cisterna per acqua piovana. La visualizzazione del livello avviene su display LCD 2x16.

Trova l'articolo di tuo interesse! Questa la legenda:

- P Progetti
- S Smart Home
- D Divulgazione scientifica

Clicca qui per scoprire chi lavora con Fare Elettronica entra anche tu a far parte, contattaci!

L'Editoriale

di Giovanni Di Maria

Si avvicina l'anno nuovo.

Quando l'elettricità e l'elettronica sono al servizio dell'uomo, i vantaggi che ne derivano risultano estremamente importanti, specialmente per la collettività e l'ambiente. Gli articoli di questo numero testimoniano proprio tale aspetto. In questo bimestre la rivista propone sei articoli veramente interessanti. L'Ing. Giovanni Carrera presenta il progetto ArduLoadCell, una realizzazione il cui scopo è quello di misurare le forze o le deformazioni con un sistema basato su Arduino. Segue l'articolo di uno dei nostri più attivi autori, Giuseppe La Rosa, che propone un sistema di controllo Strip LED RGB con Bluetooth o Display Touch Screen, che consente la gestione di una striscia di LED RGB tramite Smartphone Android oppure con un Display Touch Screen. Quindi, la seconda puntata dell'interessante articolo di Lucio Di Jasio con il suo "This is (not) Rocket Science - A proposito di Breathing e Performance". Il successivo articolo tratta di un argomento molto importante, ossia quello della mobilità elettrica con il sistema elettrico a 48 Volt. A seguire la prima puntata del progetto "Registratore consumi elettrici" di Grazia Ancona, con il quale si possono interpretare e valutare i consumi domestici, evitando eventuali sprechi d'energia. Conclude la rivista il nostro vulcanico Girolamo D'Orio che ha realizzato, e ci propone, un sistema elettronico per la gestione di una cisterna per la raccolta della preziosa acqua piovana, tramite Arduino. Colgo l'occasione per esprimere un caloroso ringraziamento ai lettori e agli autori, estendendo a tutti l'augurio di buone feste.

Giovanni Di Maria

Misure di forza con Arduino: progetto ArduLoadCell

di Ing. Giovanni Carrera

Scopo del progetto è di misurare forze o deformazioni con un sistema basato su Arduino. Le applicazioni sono molteplici, dalle bilance elettroniche, al condizionamento e acquisizione di altri sensori basati su ponti estensimetrici.

Introduzione

Le forze si misurano rilevando la deformazione di elementi elastici. Passando dalle micro deformazioni alle macro, l'esempio tipico è il dinamometro a molla, tipo quello usato dai pescatori. Se la molla è lineare, il suo spostamento sarà proporzionale alla forza applicata:

$$F = kx$$

dove k è la costante elastica della molla. Prima di descrivere i circuiti e il software, cercherò di introdurre il lettore alle misure di deformazione essendo questa la variabile fisica sensibile alla forza applicata. Passerò, quindi, a descrivere il circuito integrato HX711, molto diffuso ed economico perché usato per le bilance elettroniche che hanno prezzi ormai molto bassi. Per quanto riguarda il software, in internet esistono numerose librerie HX711, ma ho preferito sviluppare una mia funzione, ottimizzata anche per sistemi più veloci come Arduino Due. Ho usato un mio sistema basato su Arduino Due con un display LCD e modulo SD. Va altrettanto bene anche un Arduino Uno con shield LCD, per realizzare un sistema autonomo, o semplicemente con trasmissione seriale su PC via USB.

Misure di deformazione

Le deformazioni sono le variazioni di lunghezza che un corpo elastico subisce sotto l'azione di una forza, esse sono espresse in variazioni di lunghezza rispetto alla medesima, quindi adimensionali. Esse si misurano con trasduttori estensimetrici, chiamati semplicemente estensimetri. Qui di seguito descriverò l'estensimetro resistivo, che è un trasduttore che ha in uscita una variazione in funzione della deformazione. Esso è un dispositivo in grado di rivelare la

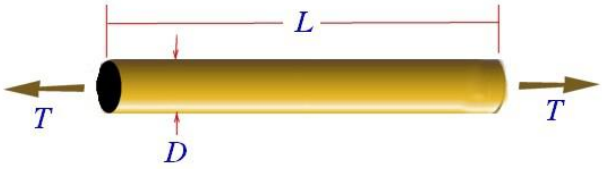


Figura 1: Tondino soggetto a trazione.

deformazione del materiale su cui è applicato mediante incollaggio. L'estensimetro resistivo elettrico permette di misurare variazioni di lunghezza dell'ordine del µm. Si consideri un sottile filo metallico di lunghezza L e diametro D, sottoposto a un carico assiale T, come mostrato in figura 1. Soggetto a trazione, il filo si deformerà allungandosi e diminuendo la sua sezione A. La deformazione assiale ε si ricava dalla legge di Hooke:

$$\varepsilon = \frac{T}{A} \cdot \frac{1}{E} = \frac{\sigma_a}{E}$$

dove σ_a è lo sforzo assiale ed E è il modulo di Young o costante elastica del materiale, espresse in pascal (N/m2) nel Sistema Internazionale. Per i metalli queste deformazioni sono dell'ordine di qualche centinaio di µm/m, quindi si esprimono in micro epsilon. Il funzionamento dell'estensimetro resistivo è basato sul fatto che la resistenza elettrica del filo è proporzionale alla lunghezza e inversamente alla sua sezione secondo l'espressione:

$$R = \rho L / A$$

dove ρ = resistività del materiale in [Ω*m]. Omettendo tutte le dimostrazioni e i passaggi matematici, si arriva all'espressione:

$$\varepsilon = \frac{1}{F} \cdot \frac{\Delta R}{R}$$

Dove F è il gage factor, una costante che dipende dalle caratteristiche piezoresistive del materiale con cui è realizzato l'estensimetro, per i metalli F≈2 ed è sempre specificato dal costruttore dell'estensimetro. In pratica, è necessario avere la resistenza dell'estensimetro molto più grande di quella dei fili di collegamento, valori tipici sono 120 e 350 Ω. Per questo occorre usare un diametro D molto piccolo e una lunghezza abbastanza grande, per cui si realizza un serpentino del filo per ridurre le dimensioni del sensore. In figura 2 si vede l'aspetto di un estensimetro a filo (a sinistra) e quello di un estensimetro a strato metallico, ottenuto per fotoincisione, è anche indicato l'asse di sensibilità. Il supporto è sempre un sottile foglio isolante, elastico, resistente alla temperatura e incollabile con colle ciano-acriliche o epossidiche.

Il ponte estensimetrico

Esistono vari tipi di celle di carico, quella che sarà usata è del tipo a flessione e a ponte intero attivo, ossia con quattro estensimetri, due che lavorano a trazione e due a compressione. Esse sono inserite in un ponte di Wheatstone, come visibile in figura 3. Il ponte di misura è ancora oggi lo strumento più idoneo per misurare con grande precisione resistenze, capacità e induttanze o loro variazioni. Per capacità e induttanze esso deve essere necessariamente eccitato in corrente alternata, per le resistenze può lavorare anche in corrente continua, come nel nostro caso. In questa configurazione tutti e quattro gli elementi del ponte sono attivi. Nel caso di estensimetri eguali con deviazioni di uguale ampiezza e a due a due opposte sui due lati del ponte, si ricava:

$$V_{bd} = E \cdot \frac{\Delta R}{R}$$

esso ha una sensibilità quadrupla rispetto alla condizione del quarto di ponte attivo, cioè con un solo estensimetro attivo e con tre resistori fissi per i tre lati. A differenza del quarto di ponte, essa è perfettamente lineare con la variazione di resistenza. Sono eliminati anche gli effetti delle resistenze dei fili e la dilatazione termica del materiale della cella. Alcune celle possono essere costruite a mezzo ponte, con due soli estensimetri attivi. In questo caso la sensibilità è dimezzata rispetto al ponte intero e si devono usare due resitori fissi di alta stabilità con la temperatura e bassa tolleranza, per completare il ponte. Il ponte deve essere alimentato da un generatore di tensione E che deve avere un'alta stabilità termica e basso rumore poiché l'amplificatore amplifica anche questi. Il guadagno del ponte è definito come il rapporto dello sbilanciamento corrispondente alla massima variazione di R e la tensione di eccitazione E:

Guadagno del ponte = $G = \frac{V_{bd}}{E}$ espresso in [mV/V]

Questo parametro è spesso usato dai costruttori di celle di carico, con riferimento al fondo-scala del sistema, un tipico valore è di 2 mV/V, ossia la tensione in uscita Vbd è pari a 2 mV per ogni volt di eccitazione del ponte Vp.

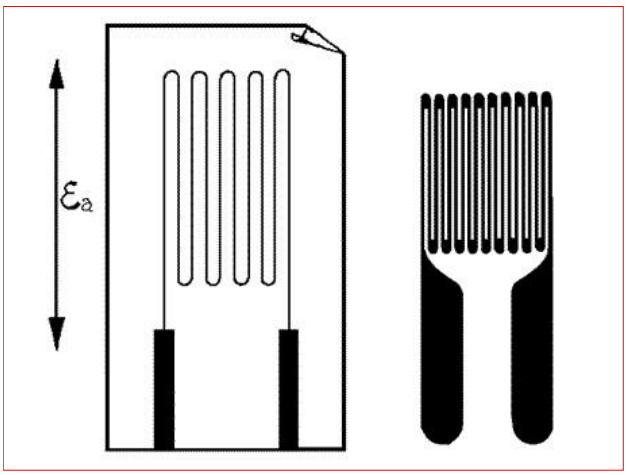


Figura 2: Estensimetri resistivi.

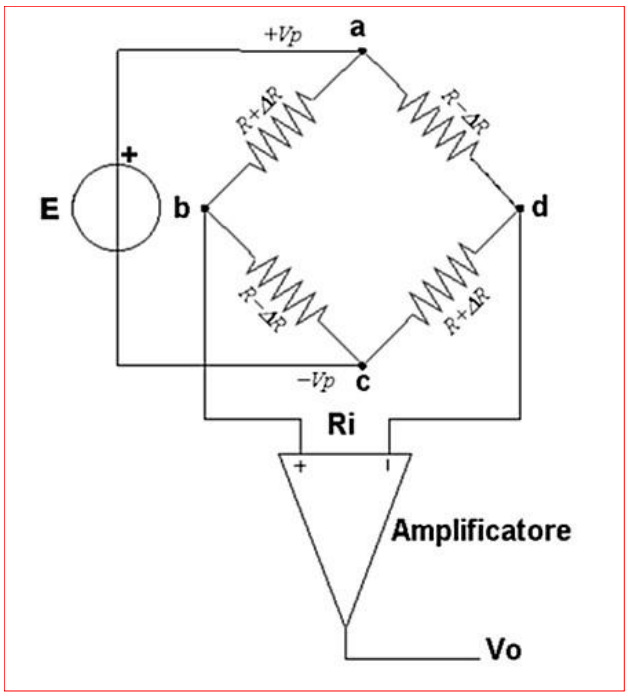


Figura 3: Ponte estensimetrico completo.

La cella di carico

Io ho usato una cella di recupero da una vecchia bilancia digitale. Sul mercato sono disponibili, ormai a prezzi molto accessibili, molte celle di carico da pochi kilogrammi forza fino a qualche centinaia. Per portate superiori i prezzi salgono notevolmente. Ricordo che, nel Sistema Internazionale, le forze sono espresse in newton [N] e non in kilogrammi forza. Il peso è una forza e non andrebbe espresso in kilogrammi che è l'unità di misura della massa. Le bilance dovrebbero indicare i newton e non la massa in kg. Com'è noto dalla fisica, a forza è

F=M*a [N]

per cui occorre moltiplicare la massa per l'accelerazione di gravità 9,80665 per avere la misura in newton. In figura 4 è visibile la cella utilizzata per le prove. Il corpo della cella è in alluminio o sua lega, sono stati praticati due grandi fori da circa 20 mm e

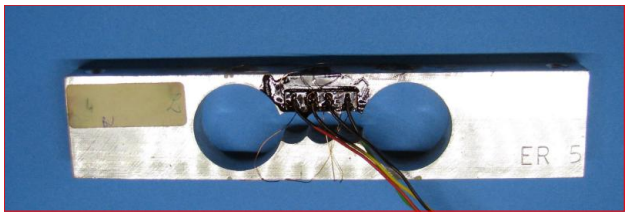


Figura 4: La cella di carico.

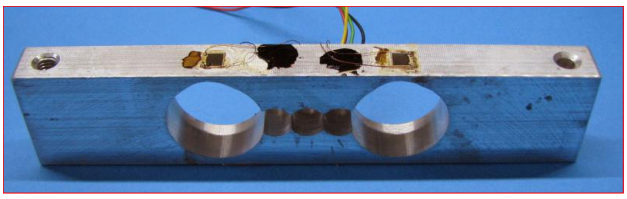


Figura 5: Disposizione degli estensimetri.

altri tre più piccoli. Le dimensioni della barretta sono: 26,2x125 mm con uno spessore di 11,2 mm. Tutti i fori servono per aumentare la deformazione e quindi la sensibilità della cella. La foto di figura 5 mostra due dei quattro estensimetri incollati in corrispondenza dell'asse dei due fori grandi. Si noti che esistono due fori filettati M4, alle estremità della barretta, uno serve per fissare la cella e l'altro per applicare il carico. Il disegno di figura 6 mostra la deformazione della cella caricata con la forza F, molto esagerata per esigenze grafiche, e le variazioni di resistenza degli estensimetri. Così com'è orientata la forza, gli estensimetri e1 ed e2 sono in trazione (ΔR positivo) mentre e3 ed e4 sono in compressione (ΔR negativo). Essi vanno collegati sul ponte come indicato in figura 3, altrimenti il ponte non si sbilancia. La cella usata era già cablata con sottilissimi fili saldati sugli estensimetri e collegati correttamente alle 4 piazzuole. A questo punto, sarebbe anche possibile auto-costruirsi la cella, anche se costerebbe di più che comprarla. Come si fa a conoscere quali dei quattro fili sono di alimentazione? Se la cella non include reti di compensazione, è facile rispondere alla domanda. Il ponte di figura 3, visto dai due vertici a-c e b-d, dovrebbe presentare, teoricamente, la stessa resistenza, che è quella degli estensimetri. Infatti sono due resistenze serie (2R) in parallelo tra loro, quindi R. Nel mio caso es-

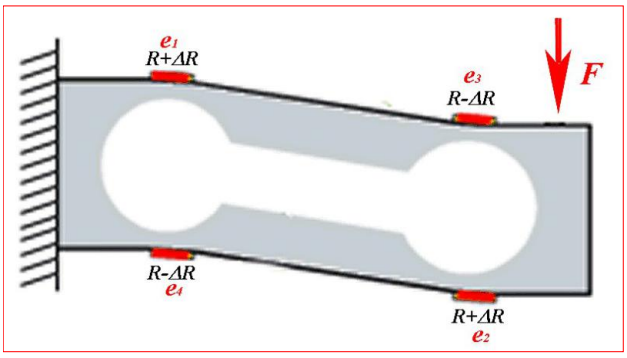


Figura 6: Deformazione della cella sotto carico.

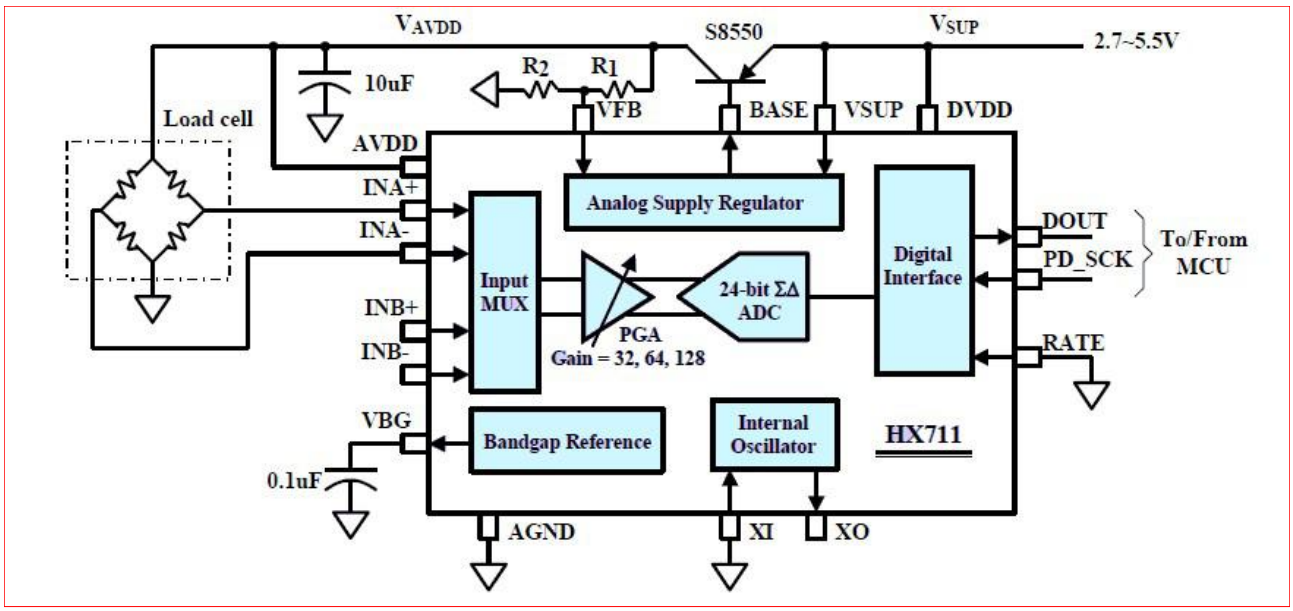


Figura 7: Schema funzionale del HX711.

sa era di 328Ω. Se misuriamo , per esempio, tra a e b, dovremmo leggere una resistenza pari a R in parallelo a 3R, quindi: R*3R/(R+3R)= 3*R/4.

Il circuito integrato HX711

Il circuito integrato è prodotto dalla Avia Semiconductor, una ditta cinese specializzata in questo tipo di convertitori, ed ha un package per montaggio superficiale del tipo SOP-16L. In figura 7 è visibile il suo schema funzionale. Partendo da sinistra, si nota un multiplex per due canali differenziali, questo perché è possibile usare due ponti A e B. Segue un amplificatore differenziale a guadagno programmabile (PGA: Programmable Gain Amplifier) a basso rumore. Per il canale A è possibile impostare i guadagni G=64 o 128 che danno un fondo scala dell'ingresso, rispettivamente di ±40 e ±20 mV per un'alimentazione di 5V, per il canale B il guadagno è fisso G=32 (±80 mV in ingresso). Segue, infine, un convertitore ADC sigma-delta (∑Δ) a ben 24 bit. Sopra è visibile l'alimentatore dei ponti, che può utilizzare un transistore esterno per aumentare la corrente in uscita. Esso ha una sorgente di riferimento di VBG = 1,25 V tipici. Per impostare la tensione in uscita, occorre il partitore formato da R1 e R2, per cui:

$$V_{AVDD}=V_{BG}*(R1+R2)/R2$$

Nella scheda usata le resistenze sono R1 = 20kΩ e R2 = 8,2kΩ, per cui la tensione di eccitazione dei ponti è pari a:

$$V_{AVDD}=1,25*(20+8,2)/8,2\approx 4,3V$$

La formula indicata nel datasheet è sbagliata, giacché la tensione sul partitore del feedback deve essere uguale alla tensione di riferimento. Un'altra importante considerazione è che la tensione sul ponte deve essere inferiore a quella di alimentazione, questo perché il transistore PNP non deve saturare per rimanere nel campo lineare. Nel nostro caso la Vce = 0,7V, un po' al limite, per cui in transistore deve avere una tensione di saturazione molto bassa. Nelle specifiche dell'integrato è dichiarato che esso può lavorare con tensioni comprese tra 2,6 e 5,5 V, per chi volesse alimentare il circuito con tensioni più basse, per esempio a 3,3 V, occorre modificare il partitore, perché la tensione di riferimento rimane sempre a 1.25V, per cui, mantenendo R2 = 8,2kΩ e volendo una V_{ADD}=2,5 V, si ottiene un partitore che dimezza, ossia R1 = R2. Si deve sostituire solo la R1, cosa non semplice con resistori smd. Il sistema HX711 ha un'interfaccia seriale sincrona realizzata con due soli bit: un segnale DOUT in uscita e un clock PD_SCK in ingresso. L'informazione

su quale canale usare e sul guadagno arriva al chip solo contando il numero d'impulsi PD_SCK con tre possibili combinazioni:

- 25 impulsi di clock impostano una prossima lettura del canale A con G=128,
- 26 impulsi del canale B con G=32,
- 27 impulsi del canale A con G=64.

Le temporizzazioni sono mostrate in figura 8, ricavata e corretta dal datasheet originale, essa presentava un errore: nel caso di 27 clock: è CHA e non CHB. Resta assodato che la configurazione di default, dopo il reset, deve essere quella del canale A con G=128 (25 impulsi). Se uno volesse il canale B (G fisso a 32), la primissima lettura è quella del canale A, e dopo il 27° impulso, la logica del chip apprende la corretta selezione per i successivi dati trasmessi. Queste considerazioni sono utili soprattutto quando si devono acquisire due canali. Il segnale PD_SCK è utilizzato anche per spegnere il chip HX711: quando è basso, il chip è in modalità di funzionamento normale. Se PD_SCK rimane alto per più di 60 microsecondi, il chip va in modalità power down, ossia si spegne e va in basso consumo. La durata degli impulsi PD_SCK (T3) deve essere compresa tra 0,2 e 50 microsecondi. In figura 8 si vedono le temporizzazioni del clock per i vari guadagni impostati. Quando DOUT è alta, il dato in uscita non è pronto per la lettura. L'ingresso PD_SCK deve essere normalmente basso, altrimenti il chip sarebbe spento. Applicando su PD_SCK da 25 a 27 impulsi di clock (a seconda del guadagno e del canale), dopo il tempo T2 (0,1 µs massimi) sono disponibili su DOUT i bit in uscita da quello più significativo (MSB), fino al bit 24 (LSB). Dal venticinquesimo impulso PD_SCK porta alto DOUT. Il dato convertito è in binario con segno, secondo la codifica complemento a 2. Utilizzando l'oscillatore interno, le conversioni possono essere generate con frequenza di 10 Hz (RATE=0) o con frequenza di 80 Hz (RATE = 1). La scheda è cablata con l'ingresso RATE (pin 15) a massa, come si vede dallo schema in figura 9. Adesso facciamo qualche studio sulla risoluzione del convertitore, sul fondo scala e sull'alimentazione stessa. Con un guadagno di 128 l'ingresso ha un range di ±20 mV, quindi l'uscita dell'amplificatore diventerebbe ±2560 mV, oltre la tensione di alimentazione di 5V. Se la cella avesse un guadagno di 2 mV/V, con circa 4,3 V di eccitazione avrebbe un fondo scala di ±8,6 mV, moltiplicati per il guadagno danno in uscita ±1100,8 mV. Per cui sarebbe preferibile usare il guadagno di 128. La conversione a 24 bit, con segno, porta a un intervallo numerico da -8388608 a +8388607. Dallo scarno datasheet non si sa esattamente quale sia l'effettivo range d'ingresso del convertitore ADC, ma, supponendo che esso sia la metà dell'alimentazione, tornano abbastanza i range in ingresso dell'amplificatore, infatti ±2500/128 ≈ ±20 mV. La risoluzione teorica di: 2500/8388607 è pari a circa 0,298 µV che, riportati all'ingresso, diventano circa 2,33 nV. Questa risoluzione non ha nessun significato fisico in quanto il rumore e le derive termiche sono maggiori di almeno due

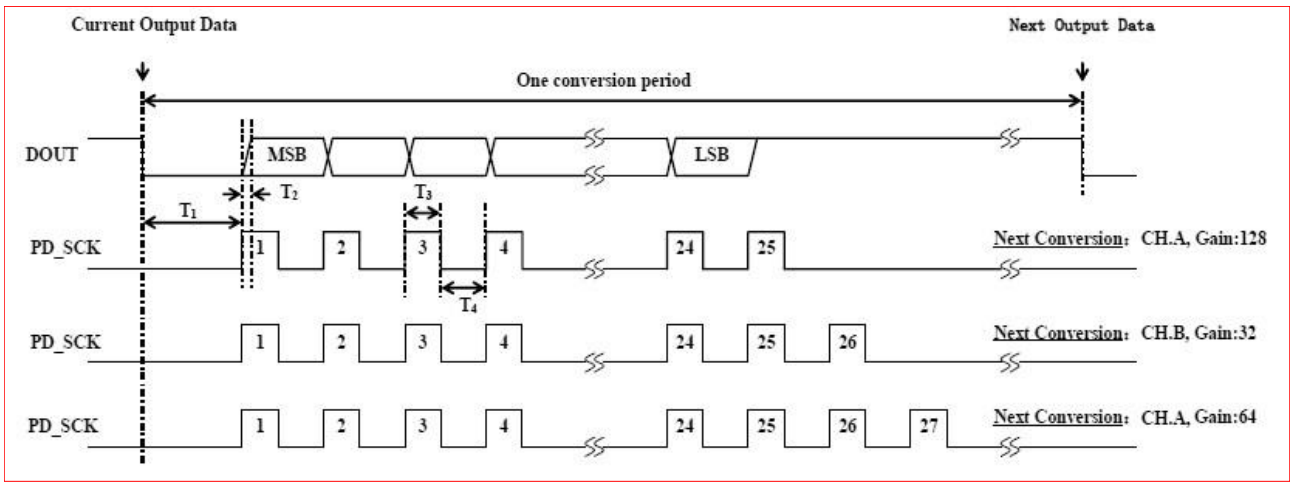


Figura 8: Temporizzazioni HX711.

ordini di grandezza. Sarebbe, quindi, più sensato ridurre il guadagno a 64 per il canale A. Per l'altro canale il guadagno è fisso a 32, la risoluzione all'ingresso è solo quattro volte superiore ed è ancora molto piccola: 9,31 nV. A questo punto dobbiamo fare un altro ragionamento. Una cella di carico professionale ha un errore complessivo tipico intorno allo 0,01% rispetto al fondo-scala, che diventa 0,05% per quelle a basso costo, come quella da me usata. Il che significa che non ha senso avere una risoluzione enormemente inferiore all'errore. Per una cella con fondo-scala di 5 kg e 0,05% di errore, ossia ±2.5 g. Basterebbe avere una risoluzione intorno alla metà dell'errore, ossia di circa un grammo. Ora bisogna calcolare la tensione corrispondente a un grammo , ossia 1/5000 del fondo-scala. Per fare questo è necessario sapere quanto è il guadagno del ponte, supponiamo che questo sia 1 mV/V (sensore poco sensibile), al fondo-scala lo sbilanciamento del ponte sarà di 4,3 mV, dividendo per 5000 ottengo la risoluzione minima richiesta, che è di 860 nV. La risoluzione del nostro convertitore è quasi cento volte più piccola sul canale B con G=32, quindi dovrei buttare via le due cifre decimali meno significative. Questo lo posso fare anche dopo aver convertito in grandezza fisica il numero.

Lo schema elettrico e la realizzazione del prototipo

In figura 9 è mostrato lo schema della scheda e dell'interfaccia con Arduino Uno e Due.

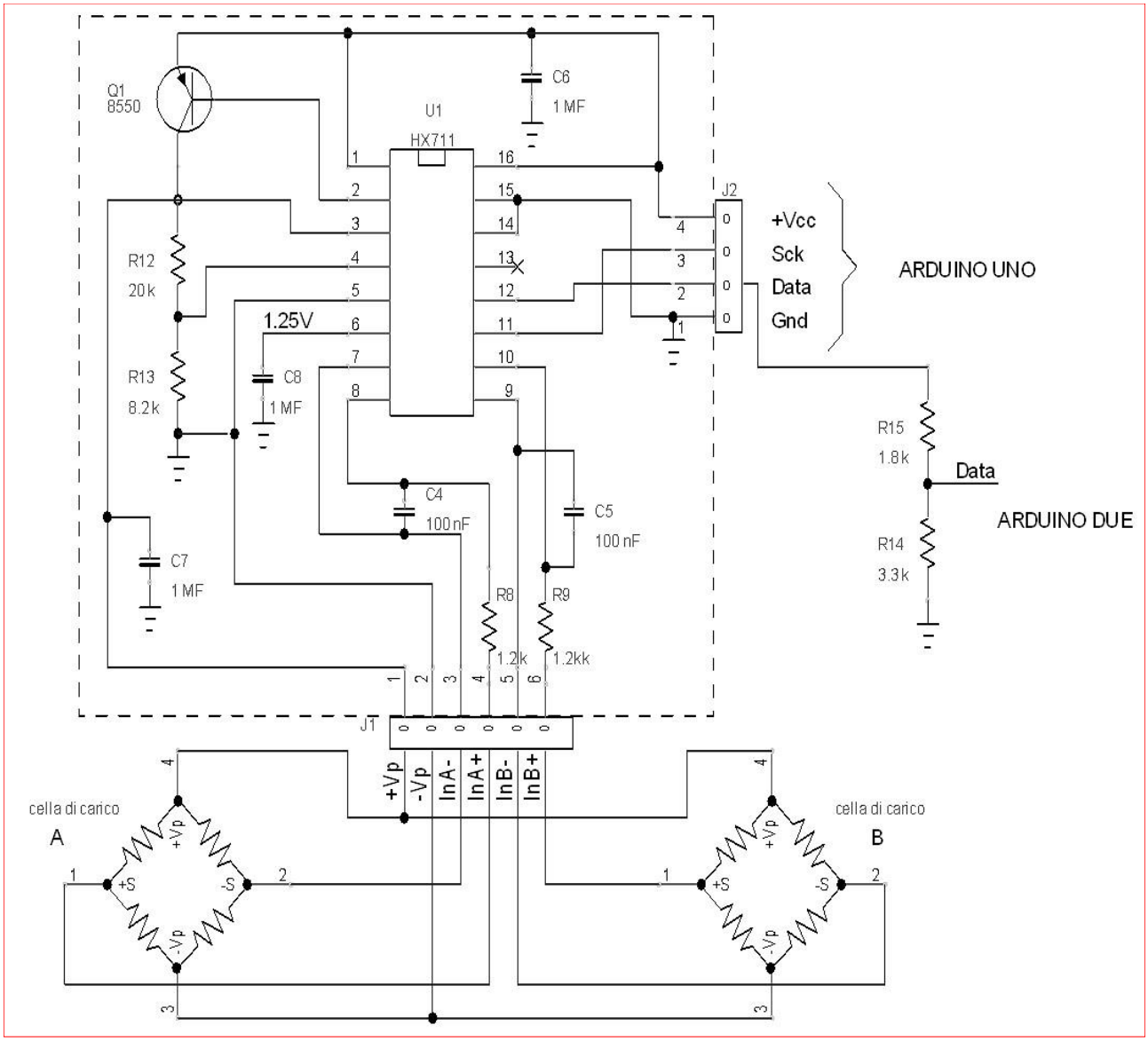


Figura 9: Schema della scheda HX711.

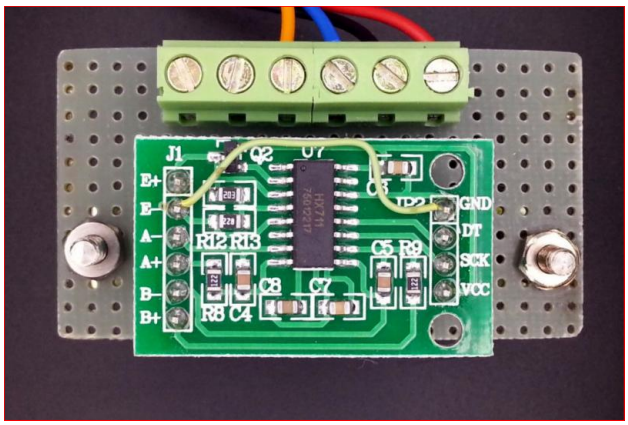


Figura 10: La realizzazione del prototipo.

Lo schema della scheda cinese che ho acquistato in internet, è stato ridisegnato per renderlo più comprensibile. I componenti sono stati contrassegnati con i numeri della serigrafia della scheda. Come descrivo più avanti, il circuito stampato della scheda cinese aveva un errore sulle masse. Essa ha due connettori a strip con passo 0,1”. Il connettore J1 serve per i collegamenti con i ponti, mentre J2 serve per l’alimentazione e per l’interfacciamento con Arduino. Poiché ho usato un Arduino Due che ha gli ingressi a 3,3V, ho dovuto aggiungere un partitore (R15 e R14) sull’uscita. Per ridurre il rumore ho aggiunto un condensatore C9, al Tantalio. I consumi della scheda sono molto piccoli, secondo il datasheet, il chip consuma solo 1,5 mA che si riducono a meno di 1 µA in power down. Il ponte consuma una corrente VAVDD/R ed è spento in power down., nel mio caso il ponte consuma: 4,3/328 = 13 mA. Come si è già detto, i due canali, per la semplicità di comunicazione adottata dal costruttore, non sono uguali come guadagno: il canale A ha un guadagno doppio o quadruplo rispetto al canale B. Forse l’uso di due ponti uguali non era prioritario per il progettista del chip, altrimenti sarebbe stato meglio usare un guadagno fisso, per entrambi i canali, con due sole serie d’impulsi per selezionare il canale.

Le prime prove

La tensione di alimentazione del ponte risultava di 3,43 V, diversa da quella che ci si poteva aspettare, cosa strana, tra la massa su J2 e –Vp del ponte ci sono ben 880 mV, questo significa che la massa analogica e quella in ingresso sono differenti, il che porterebbe l’alimentazione del ponte a 3,43+0,88 = 4.31 V, che è una tensione corretta secondo la formula del partitore. La tensione di riferimento è stata proprio di 1,25V. Facendo una misura di continuità ho riscontrato che la massa non era collegata, ciononostante il circuito funzionava lo stesso. Riguardando il datasheet, non ho riscontrato una separazione tra le masse sullo schema di figura 7, quindi, ho saldato un filo tra il pin 1 di J2 e il pin 2 di J1. Ora la tensione del ponte era di circa 4,3V, come doveva essere. La figura 10 mostra la scheda con la correzione. La schedina è collegata a una millefori mediante due connettori strip. Su quest’ultima sono saldati anche una morsettiera, il condensatore elettrolitico, il partitore e il connettore DB25P per collegarlo al mio sistema. Per le prime prove ho usato un ponte fittizio, realizzato con quattro resistenze di precisione (0,1%) e di alta stabilità (15 ppm/C°). Due resistori di calibrazione, inserite con pulsanti e deviatore, permettono di creare fino a quattro sbilanciamenti noti. Questo dispositivo è stato utile perché è molto più stabile di una cella e mi ha dato più informazioni, ma non è indispensabile, quindi non lo descrivo ulteriormente.

Misure

Per fare le prove sulla cella ho dovuto costruire un supporto, abbastanza pesante, che funge da base e un piattino leggero per appoggiare i pesi. Per proteggere i delicati estensimetri e i sottili fili ho realizzato una scocca in alluminio, che serve anche come schermatura. Ho anche saldato un cavo schermato a 4 fili. Il risultato è mostrato in figura 12, come si vede la calza metallica è collegata alla scocca. Per la calibrazione ho usato cinque pesi da bilancia da 50, 100, 200, 500 e 1000 grammi. Facendo una regressione lineare delle 6 misure (zero

pendenza [gr/N] =	0,001693
intercetta [gr] =	0,0863
correlazione=	0,9999999
errore =	0,1493335

compreso) si sono ottenuti dei risultati molto buoni, come si evince dalla tabella, dove i pesi sono in grammi-forza. Il coefficiente di correlazione è ottimo, quasi unitario, se si pensa che le misure non sono state neanche mediate. Se si vuole convertire i numeri N in grandezza fisica basta moltiplicarli per la pendenza e aggiungere l'intercetta. Come si è già detto, la risoluzione è eccessiva, in questo caso un grammo corrisponde a N = int(1/0,001693) = 591, con una risoluzione di 1.693 milligrammi. L'errore è di circa 0,15 grammi. Occorre tagliare almeno due cifre, la cosa più sensata sarebbe stampare solo l'intero, in questo caso i grammi, ecco l'istruzione da inserire:

```
dataz= float(dataz)*0.00169320+0.08632507;
```

Ovviamente, i numeri sono quelli della mia cella. Un'accuratezza maggiore si ottiene facendo una media, per esempio su 5 o 10 letture, questo riduce il rumore.

Il programma

Come già detto, dopo aver visto un paio di librerie per HX711, ho preferito realizzare una mia funzione che ho aggiunto in fondo allo sketch. La mia funzione HX711_read(GainCh), oltre ad essere più compatta e ottimizzata, prevede anche l'inserimento di un ritardo, utile nel caso si utilizzino piattaforme più veloci come Arduino Due o Teensy. Dalle prove ho verificato che ha funzionato anche senza i ritardi da 1µs. La funzione invia 24 impulsi di clock, legge i bit corrispondenti e li inserisce, a partire dal MSB, direttamente nella variabile "data", di tipo long , intero a 32 bit con segno. Questa è inizialmente azzerata per tenere nulli i bit del byte meno significativo. In questo modo si è creato un numero che è 256 volte maggiore rispetto a quello letto, ma ha lo stesso segno, per cui basta dividerlo per questo numero per ottenere il valore letto. Poi, sono trasmessi da 1 a 3 impulsi, a seconda del guadagno e del canale, come già descritto, secondo il codice GainCh, passato come argomento alla funzione. Se si vuole leggere il canale A con G=64, occorre porre GainCh=3 e fare due

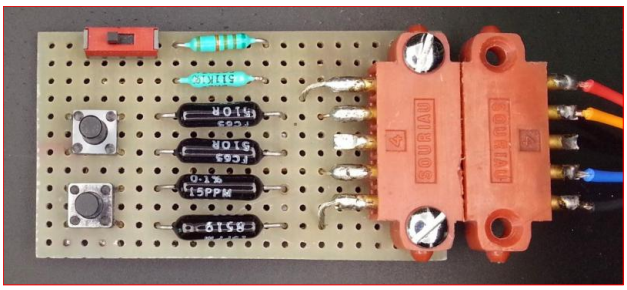


Figura 11: Il ponte fittizio.

chiamate consecutive, questo per scartare la prima che avrebbe G=128. Di Arduino ho usato solo due bit, A0 e A1, configurati come I/O digitali. Il programma vuole essere solo un esempio d'uso del chip, esegue una prima misura che potrebbe essere lo zero iniziale del ponte o la tara della bilancia. Si ricorda che il ponte è sbilanciato sia perché gli estensimetri hanno sempre delle tolleranze sulla resistenza sia perché esiste sempre un peso o tara. Nelle misure successive, quindi, è tolto lo zero iniziale per presentare i dati corrispondenti alle deviazioni del ponte soggetto a deformazioni. Con semplici modifiche si possono mediare le misure per ridurre il rumore. Non ho volutamente convertito i dati in grandezza fisica, ma questo è facilmente realizzabile mediante una calibrazione della cella con pesi noti, per esempio quelli da bilancia.

Bibliografia

1. Dispense di "Condotta Automatica degli Impianti Navali" - Capitolo 7, Giovanni Carrera.
2. "HX711 24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales", Avia Semiconductor.
3. Libreria "HX711", di Weihong Guan, https://github.com/aguegu/Arduino

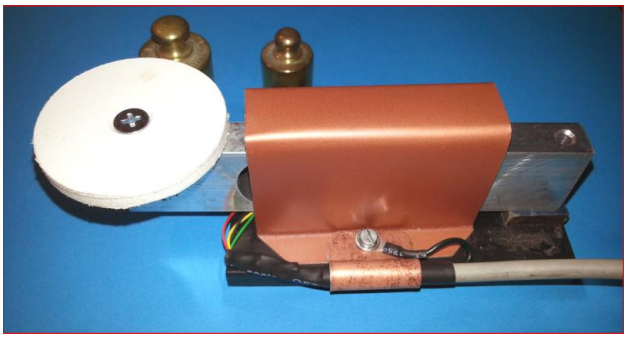


Figura 12: La calibrazione della cella.

Listato del Programma

```
/*Program HX711_LCD
 This program shows how to display analog samples from HX711 system
 GainCh = 1 for G=128 and CHA
 GainCh = 2 for G=32 and CHB
 GainCh = 3 for G=64 and CHA
 Giovanni Carrera - rev.  16 sept 2015
*/

#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(41, 40, 36, 37, 38, 39);

char bline[21] = "                    ";// blank line
int PD_SCK = A0;//pin A0 = Power Down and Serial Clock Input Pin
int DOUT = A1;//pin A1 = Serial Data Output Pin
long datai;

void setup() {
  pinMode(PD_SCK, OUTPUT);
  pinMode(DOUT, INPUT);
   // set up the LCD's number of columns and rows:
  lcd.begin(20, 4);
  lcd.print("GCAR HX711 test");// Print a message to the LCD.
  digitalWrite(PD_SCK, LOW);
  datai = HX711_read(1);// G=128 and CHA, the initial reading
  String srawdata = String(datai, DEC); // string conversion
  lcd.setCursor(0, 1);
  lcd.print("Di = "+ srawdata);// print the raw reading from the ADC
}

void loop() {
  lcd.setCursor(0, 2);
  lcd.print(bline);//print a line of blanks
  lcd.setCursor(0, 2);
  long dataz= HX711_read(1)- datai;//G=128 and CHA minus initial zero
  String scadata = String(dataz, DEC); // string conversion
  lcd.print("Ds = "+ scadata);//print  sample
  digitalWrite(PD_SCK, HIGH);    // ADC power down
  delay(3000);
  digitalWrite(PD_SCK, LOW);// ADC power on
}// end of main loop

//Read data converted by HX711 and set Gain and channel
long HX711_read(byte GainCh) {
  while (digitalRead(DOUT) != LOW);// wait low state for the chip ready
  long data = 0; // necessary for zeroing the LS byte
  // pulse the clock pin 24 times to read the data
  for (byte j = 31; j > 7; j--) {
    digitalWrite(PD_SCK, HIGH);
    delayMicroseconds(1);
    bitWrite(data, j, digitalRead(DOUT));// put the 24 bits in data
    digitalWrite(PD_SCK, LOW);
  }
  // from 1 to 3 more clock pulses for setting the channel & gain of the next reading
  for (byte i = 0; i < GainCh; i++) {
    digitalWrite(PD_SCK, HIGH);
    delayMicroseconds(1);
    digitalWrite(PD_SCK, LOW);
  }
  return data/256; // shift 8 bits right but the sign remain the same
}
```

Start-up e outcomer:

nuove idee pronte al decollo!

Preparati a entrare nell'Assodel Foundation per comunicare con il mondo dell'innovazione e conoscere i clienti del futuro...

L'Assodel Foundation

è il network di imprese e professionisti promosso dalla Federazione Distretti Elettronica Italia per supportare le start-up / outcomer a elevato contenuto tecnologico nel loro processo di crescita e di sviluppo sul mercato.

PROMOSSO DA

assodel
Federazione Distretti Elettronica - Italia

SUPPORTER

ANALOG DEVICES · ARROW · AVICEL · BETA ELECTRONICS · ISMsys · MOUSER ELECTRONICS · SGE - SYSCOM · UL

PARTNER

IngDan · Shin

outcomer@tecnoimprese.it - Tel. +39 02 210111.247

startup.assodel.it

CONTROLLO STRIP LED RGB CON BLUETOOTH O DISPLAY TOUCH SCREEN

di La Rosa Giuseppe

Questo progetto consente la gestione di una striscia di LED RGB tramite Smartphone Android oppure con un Display Touch Screen.

La scheda presentata in questo articolo (figura 1) è stata sviluppata per applicazioni con Strip LED RGB, essa è personalizzabile sia dal Hardware che del Firmware, è basa su la piattaforma Arduino UNO. La scheda consente di pilotare degli Strip LED RGB ad Anodo comune (positivo in comune) con tensione di lavoro 12 V con una corrente massima assorbita di 4 A, la scheda può essere gestita tramite un Display Touch Screen uLCD-43DT della 4D Systems oppure mediante il modulo Bluetooth HC-06 e apposita App scritta per Smartphone Android. La scheda non permette di utilizzare contemporaneamente il Display Touch Screen e il modulo Bluetooth perché sfruttano la stessa porta Seriale del Microcontrollore ATMEGA328P. Di conseguenza sono stati scritti tre appositi Firmware per l'utilizzo del Display Touch Screen e del modulo Bluetooth più uno che realizza la dissolvenza dei colori, i tre Firmware saranno approfonditi più avanti, mentre nel prossimo paragrafo analizzeremo il circuito elettrico e il suo funzionamento.

Il circuito elettrico

In figura 2 è riportato lo schema elettrico della scheda di controllo degli Strip LED RGB. I tre Mosfet Q1, Q2, Q3, sono pilotati dal segnale PWM generato dal microcontrollore IC2 (piedini IO9, MOSI, SS) essi permettono di pilotare grossi carichi come gli Strip LED RGB e consentono di variare l'intensità luminosa dei tre canali Red, Green, Blue, e di conseguenza di variare il colore della luce emessa. Il diodo LED2 RGB a Catodo (terminale negativo in comune) comune connesso agli ingressi Gate dei tre Mosfet ha la funzione di monitorare l'andamento dei tre canali RGB fornendo un riscontro visivo sull'andamento dei canali. Il connettore J2 serve per la programmazione In Circuit ed è utile per caricare il Bootloader di Arduino UNO. Il connettore J1, consente invece di caricare il Firmware dall'IDE di Arduino UNO tramite convertitore Seriale USB. Al connettore J3 va connesso il modulo Bluetooth HC-06 che permette al microcontrollore IC2 di ricevere i comandi generati dall'applicazione Android

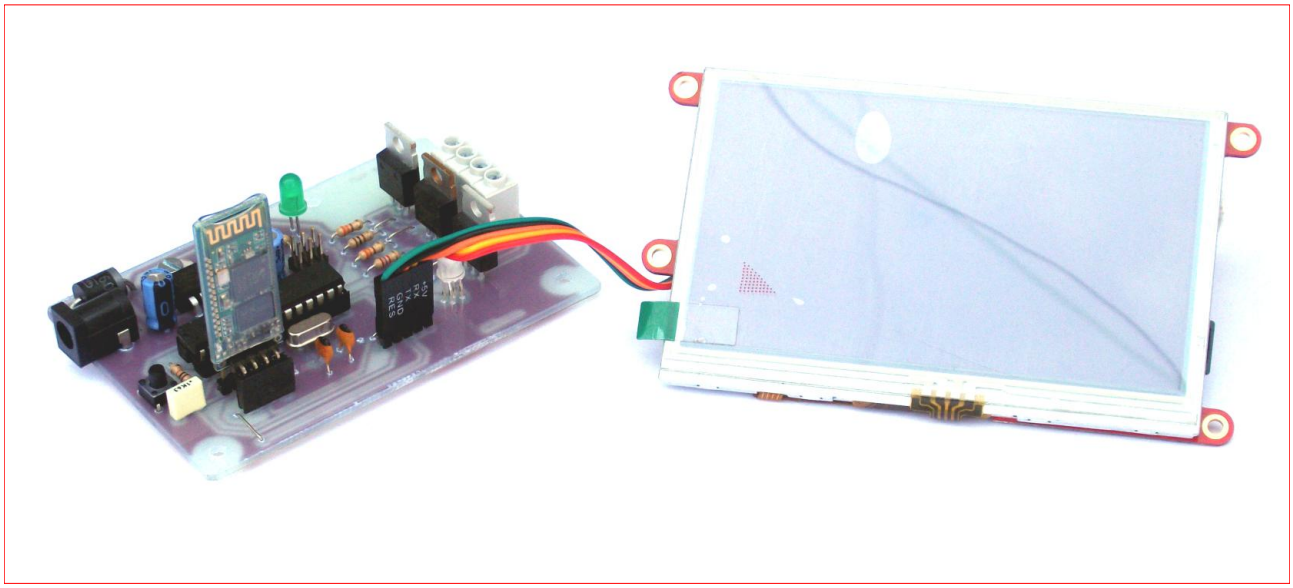


Figura 1: Foto della scheda di controllo degli Strip LED RGB.

installata su uno Smartphone. Il connettore J5 va collegato il Display Touch Screen, come è stato detto prima non è possibile utilizzare contemporaneamente il Display Touch Screen e il modulo Bluetooth perché hanno in comune la stessa porta Seriale del Microcontrollore IC2. La scheda deve essere alimentata con un alimentatore stabilizzato da 12 V in grado di erogare una corrente di 4 A, corrente sufficiente per alimentare degli Strip LED RGB di lunghezza di 5 metri. Nei prossimi paragrafi passiamo alla descrizione dei Firmware e alla programmazione del modulo Bluetooth e del Display Touch Screen.

Firmware per l'utilizzo del Display Touch Screen

Il Firmware per il controllo del Display Touch Screen è stato scritto in linguaggio C per Arduino UNO ed è possibile scaricarlo al Link che troverete a fine articolo insieme al File per programmare il Display Touch Screen. Per eseguire la compilazione o apportare modifiche al Firmware con l'IDE di Arduino UNO bisogna scaricare l'apposita Libreria fornita dalla stessa 4D Systems il cui Link per il Download lo trovate sempre alla fine. Il Firmware ha il compito di ricevere i comandi seriali provenienti dal Display Touch Screen e di conseguenza i valori dei tre Slider (Rosso, Verde, Blu) presenti nella schermata del Display vedi figura 3, essi posso variare di valore ogni uno da 0 a 255, questi tre valori prodotti dagli Slider vengono riportati direttamente sulle tre uscite PWM del microcontrollore IC2 (piedini IO9, MOSI, SS) agendo sulla variazione del valore del Duty Cycle e di conseguenza la variazione luminosa del colore degli Strip LED RGB. Il pulsante Spegni riporta le tre uscite PWM del microcontrollore al valore 0 quindi spegne gli Strip LED RGB. Passiamo alla programmazione del Display Touch Screen uLCD-43DT. Prima operazione da fare è installare il Software WORKSHOP4, Link per Download a fine articolo. Eseguita l'installazione del Software WORKSHOP4, aprite il File "VisiRGBCont" in precedenza scaricato, una volta aperto si presenterà la schermata di figura 4 connettete il Display

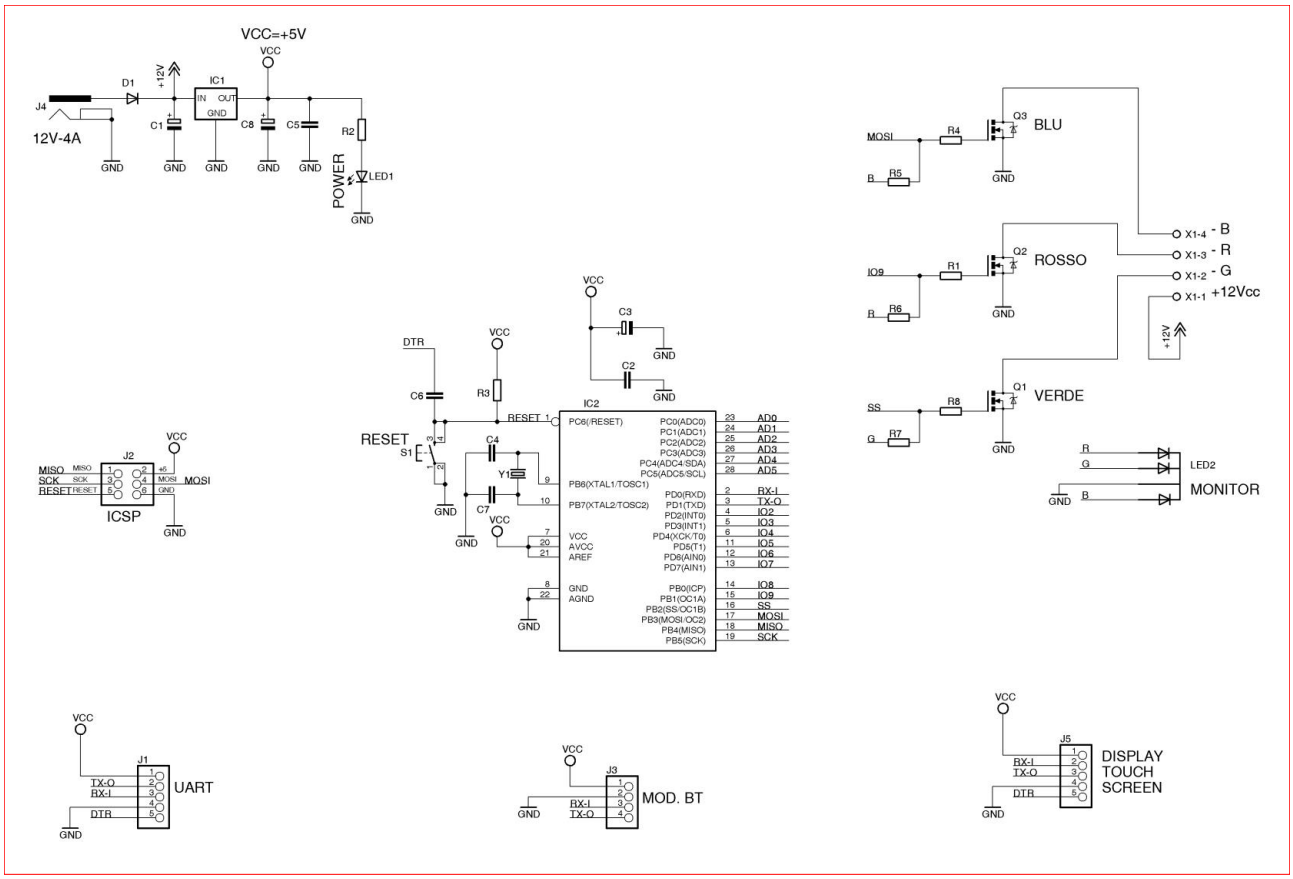


Figura 2: Schema elettrico della scheda di controllo degli Strip LED RGB.

Figura 3: Schermata del Display Touch Screen.

Touch Screen al PC tramite il convertitore USB a Seriale (con il cavetto che trovate in dotazione) della 4D Systems, poi inserite una Micro SD Card da 2 Gb nel vostro lettore di Micro SD Card e connetterlo al vostro PC, come rappresentato in figura 5. Effettuati tutti i collegamenti appena descritti basta cliccare sul tasto "Build Copy/Load" in alto a sinistra che inizierà la programmazione del Display, in seguito si aprirà una finestra che vi chiederà di selezione l'unita dove è presente la Micro SD Card eseguite la selezione dell'unita (ad esempio F) e cliccate "OK", sulla barra bianca sottostante appariranno le scritte "0 erroros" e "0 warings" ciò vuol dire che la programmazione è finita ed è andata a buon fine. Potete scollegare il Display Touch Screen dal PC ed inserire la Micro SD Card nell'apposito Slot dietro al Display, in fine connettere il Display alla scheda di controllo. Sia il Firmware Arduino UNO e il File del Display sono facilmente modificabili quindi potrete apportare tutte le variazioni e personalizzazioni che si desiderano. Nel prossimo paragrafo descriveremo il funzionamento del Firmware per poter controllare gli Strip LED RGB tramite Smartphone.

Firmware per l'utilizzo del modulo Bluetooth HC-06

Il Firmware riconosce la sequenza di valori numerici (inviati dai tre Slider) inviati dall'applicazione Android (vedi figura 6) e provvede a modificare la luminosità dei tre colori degli Strip LED RGB. Il secondo pulsante dell'applicazione Android posto in alto provvede a spegnere gli Strip LED

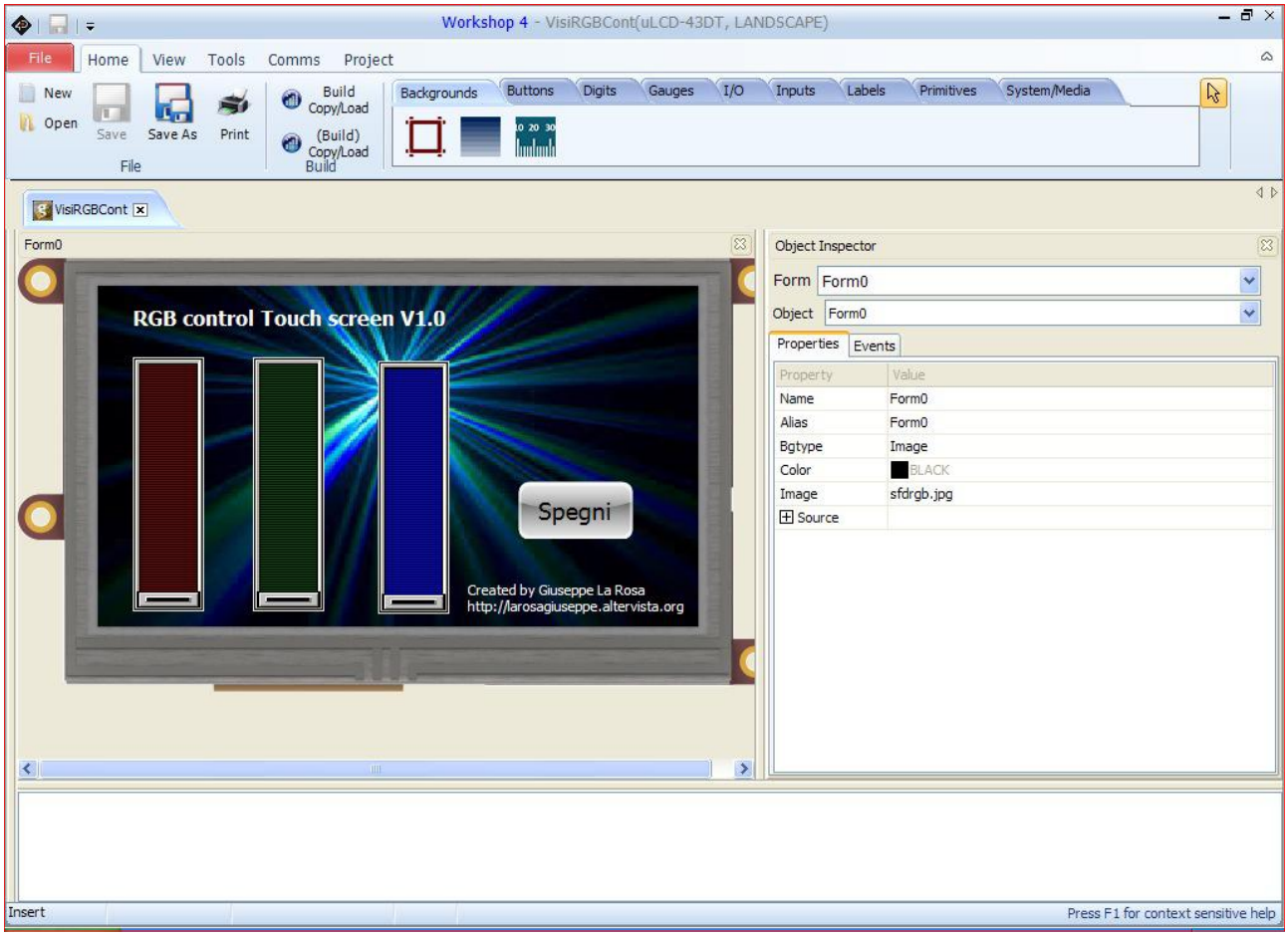


Figura 4: Schermata del Software ViSi Genie per la programmazione del Display Touch Screen.

RGB, come è facile intuire è stato mantenuto lo stesso modo di funzionamento usato per il Display. È stato solo aggiunto il primo pulsante in alto che consente di connettere e di scegliere il dispositivo (scheda di controllo RGB) da utilizzare. Data la semplicità e i molteplici commenti inseriti nel sorgente del Firmware non ci dilunghiamo nella sua descrizione, vi consigliamo di leggerlo in autonomia. Ora passiamo alla programmazione del Modulo Bluetooth HC-06. Esso supporta i comandi AT per modificare velocità di trasmissione, nome dispositivo, Passkey. Ha un settaggio predefinito della porta Seriale "8-N-1 9600" cioè dati a 8 bit, nessuna parità, 1 bit di stop con velocità 9600 Bps. La Passkey predefinita di accoppiamento è la seguente "1234". Per il nostro utilizzo imposteremo la porta Seriale "8-N-1 19200" la Passkey "1234" e il nome come "StripRGB". Avvalendoci di convertitore USB a Seriale (ad esempio quello della 4D Systems), ed eseguendo i collegamenti al modulo HC-06 come illustrato in figura 5 possiamo modificare tutti i parametri appena descritti. Utilizzando il software Hercules vedi figura 7, (Link per il Download a fine articolo) andremo ad inviare per prima il comando "AT" che ci serve a verificare che il collegamento e la porta Seriale siano stati selezionati correttamente. Se il modulo risponde "OK" significa che tutto è stato connesso correttamente e possiamo proseguire con l'invio degli altri comandi. Inviamo il comando per cambiare il nome del dispositivo con "AT+NAMEStripRGB", poi il comando per cambiare la Passkey "AT+PIN1234", ed infine il comando per cambiare la velocità della porta a 19200 Bps con il comando "AT+BAUD5". Con questa semplice procedura abbiamo settato il modulo e adesso è pronto per essere utilizzato con la scheda di controllo Strip LED RGB. Abbiamo realizzato un altro Firmware molto semplice che realizza la dissolvenza dei colori automatica e non ha bisogno di Display o modulo Bluetooth, lasciamo al lettore l'analisi del funzionamento e le eventuali modifiche. Nel prossimo paragrafo passiamo alla parte pratica cioè alla realizzazione della scheda.

Elenco componenti

R1	330 Ω 1/4 W
R2	470 Ω 1/4 W
R3	10 KΩ 1/4 W
R4	330 Ω 1/4 W
R5	100 Ω 1/4 W
R6	180 Ω 1/4 W
R7	100 Ω 1/4 W
R8	330 Ω 1/4 W
C1	100 µF 35 V elettrolitico
C2÷C3	100 nF poliestere
C4	22 pF ceramico
C5÷C6	100 nF poliestere
C7	22 pF ceramico
C8	100 µF 35 V elettrolitico
D1	6A60 diodo
Q1÷Q3	BUK9535
IC1	L7805CV
IC2	ATMEGA328P
LED1	Led 5 mm Verde
LED2	LED 5 mm RGB c.c.
Y1	Quarzo 16 MHz
S1	Pulsante c.s.
J1	Strip maschio 5 pin
J2	Strip maschio 3+3 pin
J3	Strip femmina 4 poli
J4	Presa DC 90° 5,5x2,1 mm
J5	Strip maschio 5 poli
X1	Morsetto 4 poli
P1÷P6	Ponticello
N.1	Modulo Bluetooth HC-06
N.1	Display Touch Screen uLCD-43DT
N.1	Micro SD Card da 2 Gb
N.1	Zoccolo 14+14 pin
N.1	Vite 3x15 mm più dado

Realizzazione delle schede e collaudo

Passiamo adesso alla costruzione della scheda che si presenta abbastanza semplice, la basetta è del tipo mono faccia e si prepara a partire dalla traccia di figura 8. Ottenuto il circuito stampato, iniziate a montare la scheda (seguendo il piano di

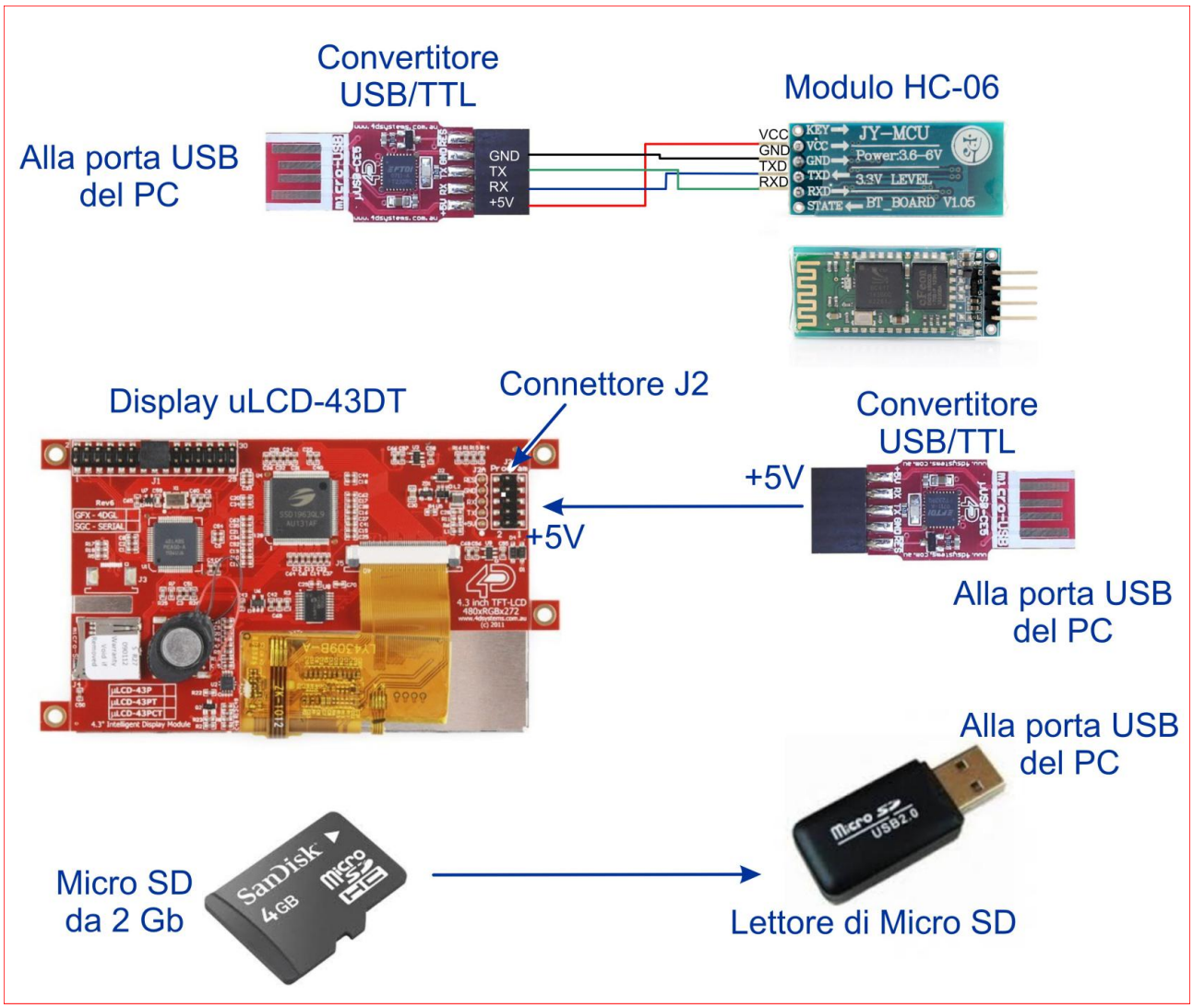


Figura 5: Schemi di collegamento per la programmazione del modulo Bluetooth e Display Touch Screen.

montaggio figura 8) e i componenti richiesti dall'"Elenco componenti". Per prima cosa eseguite tutti i ponticelli che vanno da P1 a P6. Inserite tutte le resistenze, per ultimo il diodo D1 che va saldato a 5 mm dalla piastra. Saldate lo zoccolo per l'integrato IC2, proseguite con i condensatori non polarizzati e poi gli elettrolitici, il quarzo Y1, gli STRIP, i LED, e i connettori X1 e J4, per ultimi i tre il Mosfet Q1 a Q3. Se si dispone di un ATMEGA328P già con Bootloader caricato si può passare al caricamento del Firmware da voi scelto tramite un convertitore USB a Seriale che va collegato al connettore J1 presente sulla scheda. Altrimenti prima bisogna caricare tramite un programmatore il Bootloader dal connettore J2. Eseguite il collegamento come è rappresentato in figura 9, in base al Firmware prescelto in precedenza realizzate il collegamento della configurazione che intendete adoperare o con il Display Touch Screen oppure con il modulo Bluetooth. Usate un alimentatore stabilizzato a 12 V in grado di erogare una corrente continua di almeno 4 A, se intendete usare degli Strip LED RGB in più aumentate la corrente di almeno 2 A.

Conclusione

Come avete letto in precedenza il controllo Strip LED RGB può essere personalizzato sia a livello di Hardware e di Firmware, quindi adattabile alle vostre esigenze, esso può essere usato per la retroilluminazione di Monitor LCD o TV, oppure per illuminazione scenografica di ambienti. Di seguito trovate tutti i Link per scaricare i File per la realizzazione del progetto, ma anche i Link dove trovare il materiale per la realizzazione.

Figura 6: Screenshot dell'App Android per il controllo tramite Bluetooth e QR Code per il download.

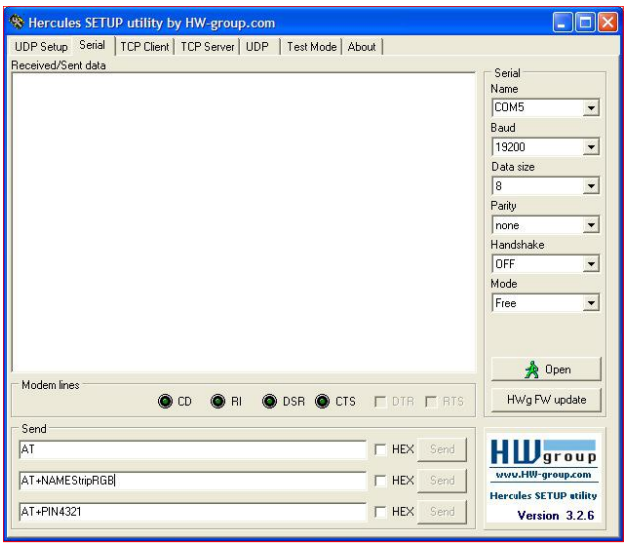
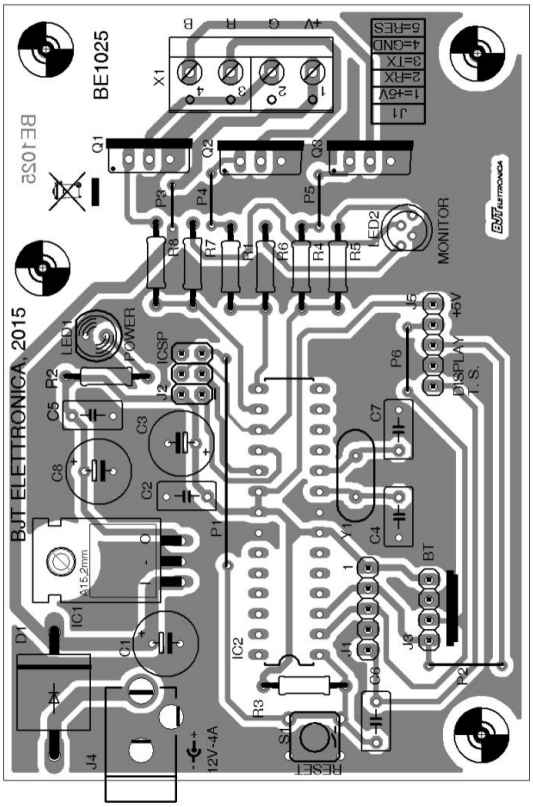


Figura 7: Schermata del Software Hercules per programmazione del modulo Bluetooth HC-06.



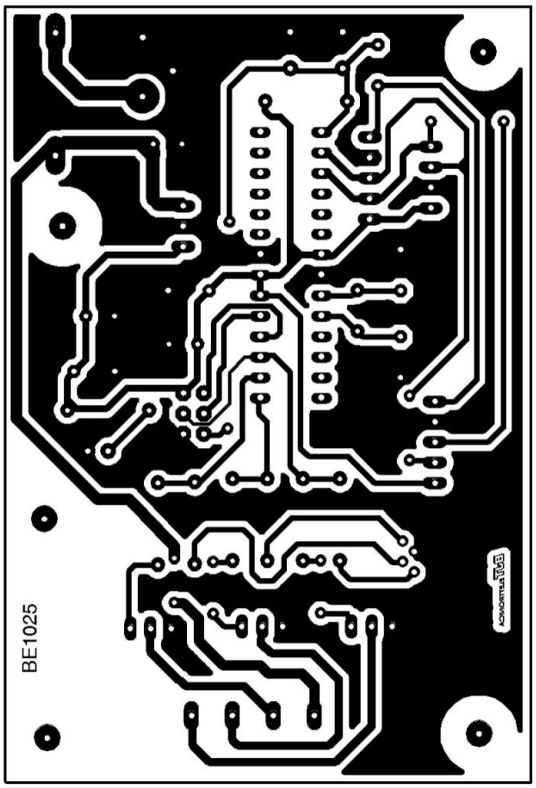


Figura 8: A sinistra piano di montaggio a destra master del circuito stampato della scheda di controllo degli Strip LED RGB.

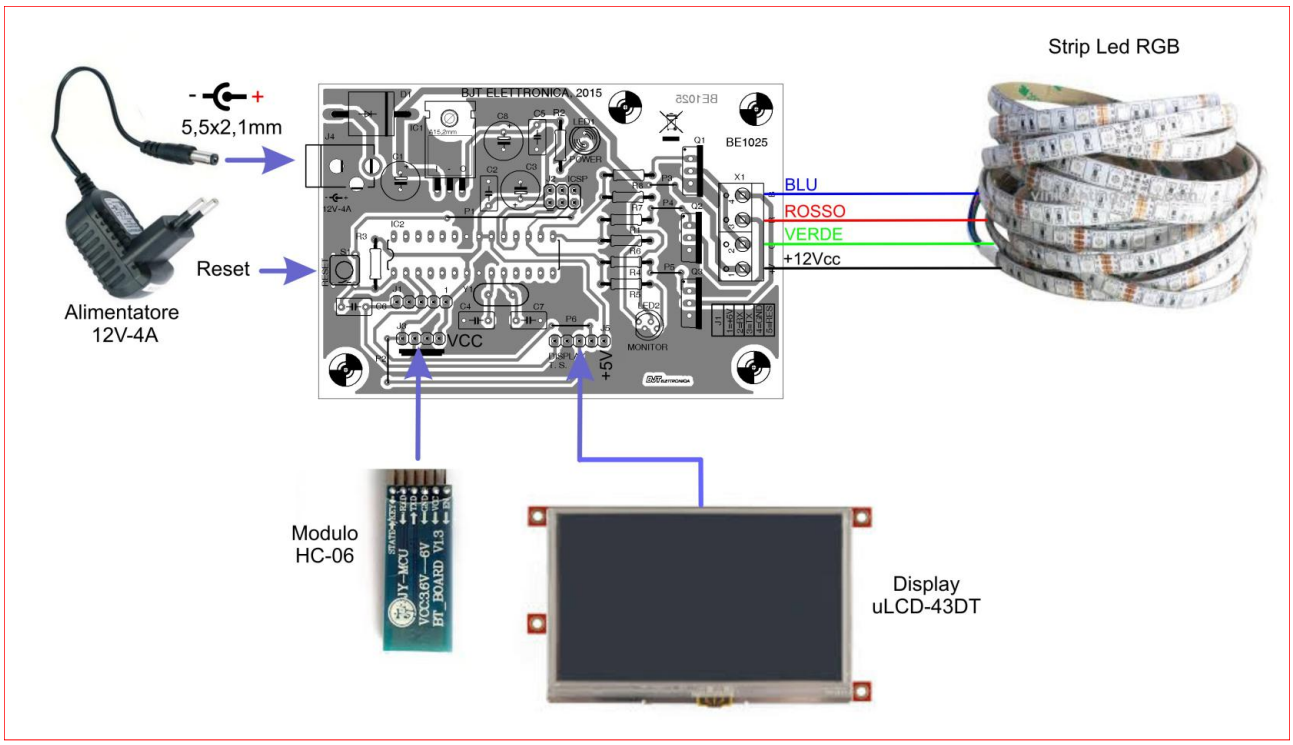


Figura 9: Schema di collegamento tra la scheda e vari dispositivi.

Riferimenti

Per scaricare il Software WORKSHOP4 e la libreria per Arduino UNO:

http://www.4dsystems.com.au/product/4D_Workshop_4_IDE/

Se volete acquistare il Display uLCD-43DT:

http://www.elettroshop.com/ulcd-43dt-diablo16-display-intelligente-tft-lcd-da-4-3-con-touch-resistivo/

Per scaricare il sorgente per l'ATMEGA328P e il File ViSi Genie:

http://larosagiuseppe.altervista.org/FwBE1025.rar

Per scaricare l'APP per gli Smartphone Android:

http://larosagiuseppe.altervista.org/RGBcontrol1V0.apk

Per scaricare il Software Hercules per la programmazione del modulo HC-06:

http://new.hwg.cz/files/download/sw/version/hercules_3-2-8.exe

Guardare il video del progetto in funzione:

https://www.youtube.com/watch?v=oeYi9OH_kGM

THIS IS (NOT) ROCKET SCIENCE

Parte 2: A proposito di Breathing e Performance

di Lucio Di Jasio

Non molto tempo fa ho partecipato ad un hands on workshop che aveva l'obiettivo di introdurre un gruppo di "millennials", che già aveva una base di conoscenza di informatica, all' arte della programmazione embedded. Come previsto, uno dei primi progetti esempio ha finito per essere il proverbiale "Hello Embedded World" che, naturalmente, significava far lampeggiare un LED. L'istruttore aveva fatto realmente un ottimo lavoro nell'evitare di essere sviati dai tanti dettagli e tentazioni. Aveva guidato la classe attraverso la realizzazione di una semplice porta I/O ed era arrivato dritto all'obiettivo utilizzando soluzioni tra le più semplici possibili. Ciò voleva dire un paio di cicli bloccanti (in attesa) e due compiti che, ci aveva spiegato, sarebbero stati certamente convertiti in singole istruzioni MCU (bit-set e bit-clear) dall'efficiente C compiler. Poco dopo, l'aula fu invasa dalla luce intermittente di tanti (piccoli) LED rossi ma, con delusione del relatore, la moltitudine non sembrava cosi entusiasta del risultato. Dal fondo dell'aula si iniziò a udire un fitto borbottare. La fonte di tanto malcontento di quel gruppo fu presto svelata: come spesso accade, era una questione di aspettative alimentate in maniera errata. C'è il blinking, e poi c'è il pulsing o breathing, come i modermi utenti Mac (e molti altri mobile) definiscono quando i loro dispositivi sono in standby. Apple, ti maledico!

A corto di MIPS

In seguito, la classe prese una svolta inaspettata quando il relatore cercò di soddisfare l'uditorio, partendo per la tangente per introdurli alle relative complessità che il nuovo task avrebbe implicato. Si dovette aggiungere un timer e con esso il PWM dovette essere adattato per regolare l'uscita luminosa visibile del LED controllandone il duty cycle. Quindi si avventurò nella spiegazione di come questa doveva essere modificata gradualmente, attraverso un numero di passaggi, aumentando e diminuendo, definiti in una tabella. Il timing di ogni step richiese ancora un altro (più lento) timer, che produceva un interrupt. L' MCU avrebbe dovuto rispondere a questo interrupt e calcolare (o consultare una lookup table) per assegnare un nuovo valore di duty cicle al PWM. Questo fu davvero troppo per l'uditorio, alla loro prima incursione nel mondo dell'embedded, ma le cose peggiorarono ancora quando cercò di impressionarli abbozzando due conti. Ecco i calcoli, come li ricordo io:

1. Scegli una frequenza PWM tale da poter contare sull'effetto di persistenza dell'occhio umano: 30~120Hz.
2. Moltiplica quel valore in modo tale da offrire un graduale effetto di attenuazione, diciamo in 256 step. Questo da ~ 32kHz. Fin qui tutto bene. C'è un oscillatore interno extreme low power disponi-

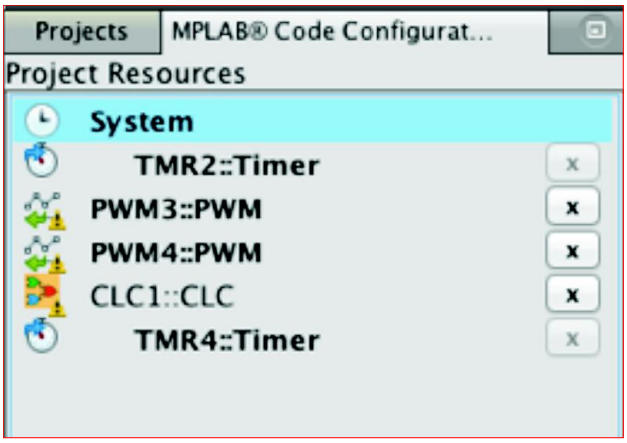


Figura 1: Finestra delle Project Resources MCC

bile all'interno di ogni microcontroller PIC, ad ha esattamente quella frequenza!

3. Ora, se vogliamo produrre il più semplice effetto breathing (utilizzando un profilo di forma triangolare) con una frequenza tra ½ secondo e 2 secondi, dovremo incrementare e diminuire 512 (in totale) step per ogni periodo.

4. Questo più o meno si traduce nella necessità di aggiornare il PWM approssimativamente ogni millisecondo. In termini di cicli MCU, alla frequenza di clock di 32kHz, dovremmo ricevere un interrupt ogni 8 cicli di istruzioni.

Persino al progettista embedded più inesperto questo suonerebbe come una proporzione impossibile. Il clock dovette quindi essere aumentato o, in parole povere, stavamo andando a corto di MIPS! Questa fu una rivelazione per alcuni, tra il pubblico, dato che per la prima volta avevano compreso che il termine performance (sia in MIPS che MHz) aveva un significato molto diverso in questo contesto. Avevamo bisogno di spingere più in alto il microcontroller sebbene non stessimo cercando di eseguire alcun calcolo, dovevamo rispondere tempestivamente ad un evento! Quello era anche il loro primo incontro con il concetto di real time performance.

Segui il Ritmo

L'esercitazione incontrò anche il mio favore, ed è la ragione per la quale ne state leggendo qui. Ho cercato modi semplici per illustrare le limitazioni del tradizionale approccio MCU/core centric verso le applicazioni embedded. Oggi siamo ossessionati dalle prestazioni, MIPs, MHz e Mega byte, e tuttavia spesso concentriamo la nostra attenzione sul tipo di performance sbagliato. A volte, fare un passo indietro e guardare il problema da una diversa e nuova angolazione, può rivelare una soluzione nascosta, più elegante e bilanciata. Per esempio, quando si approccia il caso del breathing dei LED presentato sopra tenendo la mente libera dalle catene della cultura MCU/core centrica, potrete notare una stretta similitudine tra impulsi visivi del LED e l'effetto acustico prodotto dal "battito" di due segnali ricorrenti di frequenze molto vicine, dato che esse si sommano e si elidono reciprocamente. La somma delle due raggiunge le nostre orecchie con un envelope che è riconoscibilmente ciclico e con frequenza uguale alla differenza tra le due. Non è difficile da implementare con solo una coppia di gate logici e può essere realizzato con solo un microcontroller e con il giusto tipo di periferiche!

Una Soluzione Core Independent Solution

Ciò di cui per primo abbiamo bisogno è trovare il modo per generare due segnali periodici (vanno bene le onde quadre) le cui frequenze distino solo 0,5 - 2Hz. Una coppia di timer digitali con un reload register (in altre parole periferiche PWM basiche) potranno essere utilizzate per creare i due segnali. Dovremo scegliere con attenzione i due valori reload, perchè siano vicini ma con una differenza tra i due tale da generare la desiderata frequenza di battimento. Per esempio, potremmo usare un microcontroller PIC con un

Figura 2: Finestra di configurazione di sistema MCC, dettagli sul Clock

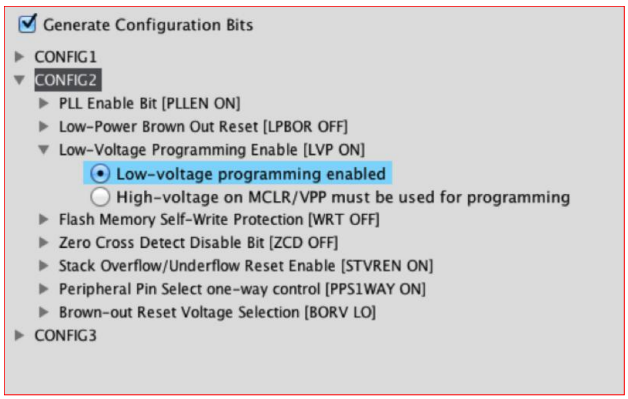


Figura 3: finestra di configurazione del sistema MCC, e dettagli Config Bits

clock a 32kHz, due timer 8-bit ed il modulo PWM connesso, e si potrà generare una frequenza di uscita pari a 30,5Hz ed una seconda di 31,5Hz. Possiamo utilizzare una delle Configurable Logic Cell (blocchi logici programmabili simili ai FPGA/PLD macro block), per eseguire la logica AND dei due segnali. Infine potremo inviare l'uscita del CLC direttamente verso qualsiasi degli I/O pin dove sia collegato un LED. Questo ci darà un visibile effetto breathing di ~1Hz. Breathing più rapidi si potrano raggiungere incrementando il gap tra le due frequenze (rendendo il periodo del secondo PWM più breve), viceversa breathing più lenti si raggiungeranno riducendo la differenza tra le due fino a ~0.1Hz quando i due reload register avranno un solo conteggio di differenza. Il Configurable Logic Cell è la più basica tra le Core Independent Peripherals (I) presenti in un moderno microcontroller (PIC), e per il resto della soluzione abbiamo utilizzato solo timer del caso piuttosto standard (PWM). Vi starete chiedendo se, a questo punto, non abbiamo più utilizzato MCU da soli! Infatti il core del microcontroller è utilizzato durante l'avviamento dell'applicazione per configurare configure le periferiche. Probabilmente ancora più interessante, abbiamo ottenuto la nostra soluzione utilizzando il nostro oscillatore dal più basso consumo di potenza e con il 100% delle prestazioni del microcontroller (MIPS o MHz non importa come volete misurarlo) disponibile per l'utilizzo in "altre", possibilmente più interessanti, task!

Mamma guarda, senza codice!

Se già così la cosa suona fantastica, vi farà ancora più felici poter imparare come tradurre ciò nella pratica dato che non sarà necessario forzare nessun datasheet, nè scrivere una singola riga di codice! Vedere per credere, seguitemi! Per semplicità utilizzerò il PICDEM Curiosity (II) board, ma una qualsiasi altra scheda in grado di ospitare un micro PIC 14/20-pin andrà bene. Ne consegue che utilizzerò un PIC16F1619, opportunamente fornito con scheda (sebbene sia dotato di socket e facilmente sostituibile). Utilizziamo quindi il wizard "New Project" dell' MPLAB X IDE per creare un nuovo progetto con il modello di PIC scelto. Utilizziamo anche l' MPLAB Code Configurator (III) (v 2.52), un plugin gratuito, per aiutarci nell'inizializzazione e per collegare tutte le periferiche. Potremo semplicemente selezionarli dalla lista di "Device Resources" con un doppio click sul nome. Nel nostro caso, faremo doppio click su PWM3, PWM4, ed uno dei moduli CLC – Io per esempio ho scelto CLC1. Nella finestra in alto, dove le "Project Resources" scelte sono elencate, ved. Figura 1, potremo ora cliccare su ognuno di loro e procedere all'esaminare le loro finestre di dialogo per l'installazione. Ed è qui che potremo saperne di più riguardo le specifiche opzioni disponibili per ognuna delle risorse. La parte superiore della lista contiene sempre le voci di "System". Ciò rappresenta gli elementi essenziali per il microcontroller quali i vari oscillatori ed i

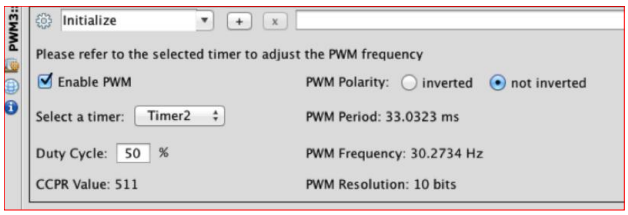


Figure 4: PWM3 configuration window

Configuration Bit. Impostate l'oscillatore nella modalità "31kHz_LF" (ha il più basso consumo, tra tutti) come in figura 2 e, poche righe sotto, ricordatevi di abilitare il bit "LVP" di configurazione come in figura 3. Ciò è richiesto per il circuito di programmazione PIC built-in della scheda per poter funzionare correttamente. Quindi cliccate sulla risorsa PWM3, ved. figura 4. Noterete che un timer è stato già selezionato come suo time base – per default questo è identificato come Timer2. Quale conseguenza, MCC lo ha già aggiunto alla lista di "Project Resources" e potete quindi procedere a configurarlo, dato che tutte le altre opzioni sono già impostate come necessarie (di default). Queste includono: un duty cycle del 50%, ed una polarità di uscita non-invertita. Cliccando su TMR2, ved. figura 6, ci viene presentato ancora una volta un numero di valori sensibili di default, compreso il valore di period già impostato su 0xFF (255). Cliccate ora su PWM4, ved. figure 5, e modificate il suo time base con Timer4, cosi che sia possibile selezionare un diverso period. Non appena lo farete, vedrete apparire TMR4 nella lista delle "Project Resources". Cliccatelo, ma questa volta modificate il valore di period con 0xf7 (247), come in figura 7. In

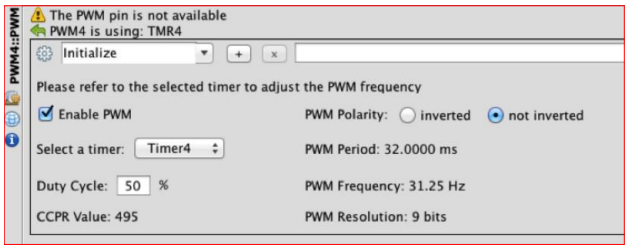


Figura 5: finestra di configurazione PWM4

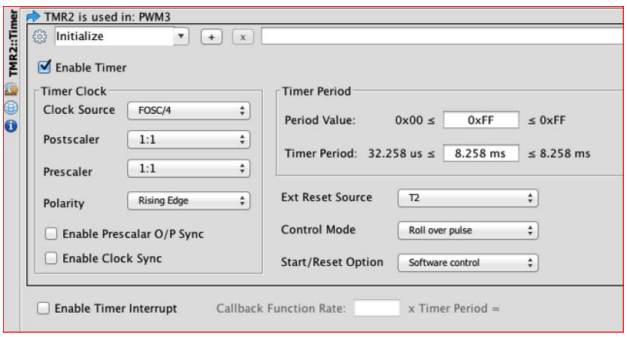


Figura 6: finestra di configurazione TMR2

ultimo, cliccate sul modulo CLC1 e configurate i primi due segnali perchè si connettano rispettivamente agli output PWM3 e PWM4, ved. figura 8. Collegateli al GATE1 e GATE2 e assicuratevi che la funzione "AND-OR" sia selezionata. Successivamente, usate il pulsante "Pin Manager" per accedere alla tabella di configurazione I/O dove dovrete assegnare un pin all'output di CLC1: RA2 nel nostro caso, che è connesso fisicamente ad uno dei quattro LED della scheda Curiosity. Premendo il pulsante "Generate Code" si attiverà l'MCC che produrrà un set di sei piccoli file sorgenti (scritti in C) che forniranno tutto il necessario codice di inizializzazione delle periferiche. Infatti l' MCC offrirà di produrre un file principale. Accetteremo ovviamente con piacere l'offerta, e questo aggiungerà un file "main.c", che conterrà le chiamate alla inizializzazione della periferica ed un empty main loop. Non siate sorpresi se, a questo punto, vi suggerisco di chiedere semplicemente all' MPLAB X di costruire il progetto e programmare la scheda Curiosity con un singolo e ultimo

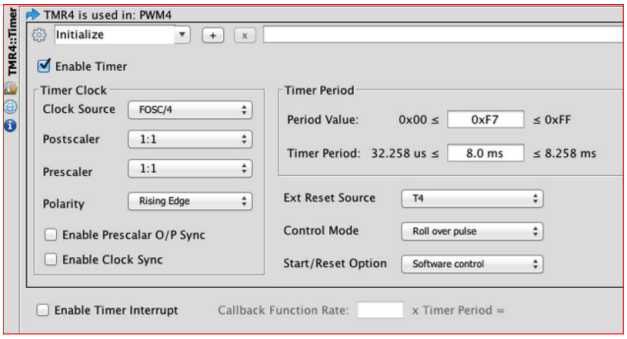


Figura 7: finestra di configirazione TMR4

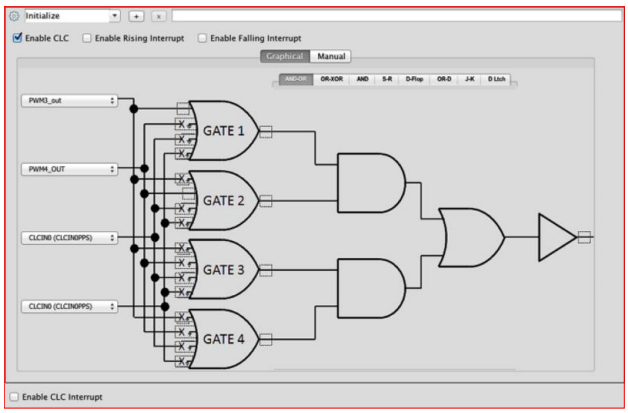


Figura 8: finestra di configurazione CLC1

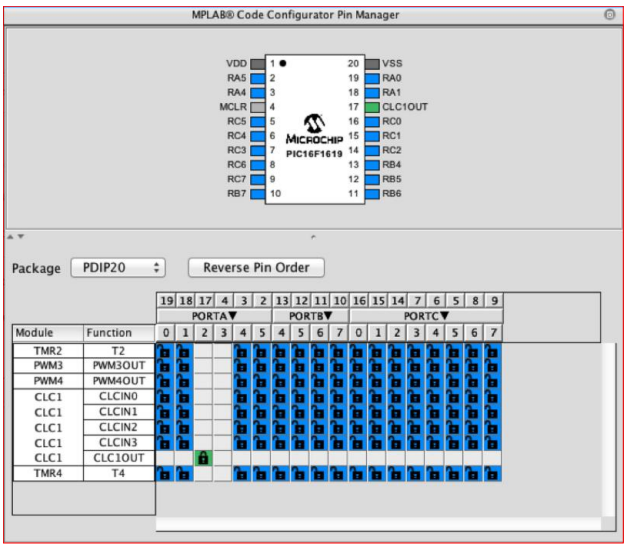


Figura 9: finestra MCC Pin Manager

click sul pulsante “Make and Program”. Dopo pochi secondi, necessari ai compiler, linker e programmer per eseguire il loro compito, sarete felici di vedere il Breathing del LED della piccola scheda Curiosity!

In meno di 10 righe di codice (binario)

Mi spiace dovervi notificare che, contrariamente alla precedente parte di questa serie (“In 10 righe di codice”), questa volta non abbiamo dovuto digitare “manualmente” una singola riga di codice! Semplicemente combinando alcune periferiche (core independent) per costruire una nuova funzione, avremo fatto e terminato e avremo ancora il 100% delle prestazioni del nostro MCU disponibili per utilizzarle nella (restante) nostra applicazione. Troverete il progetto contenente la configurazione MCC e tutto il codice generato per questo articolo all’interno dell’archivio Rocket su GitHub (IV). Altri progetti ed esempi basati sulle schede (Low Pin Count) Curiosity e Simplicity tratte dal mio blog e dal libro “This is (not) Rocket Science” (V) si possono trovare su GitHub nei seguenti archivi (VI).

Biografia

Lucio Di Jasio è l’ EMEA Business Development Manager di Microchip Technology Inc.. Ha ricoperto vari ruoli tecnici e di marketing all’interno delle divisioni 8, 16 e 32bit dell’azienda negli ultimi 18 anni. Quale autore tecnico prolifico e stimato, Lucio ha pubblicato numerosi articoli e svariati libri sulla programmazione per applicazioni di Embedded Control. Inseguendo la sua passione per il volo ha ottenuto anche le certificazioni FAA e EASA per la licenza di pilota privato. Altre notizie ed informazioni sugli ultimi libri e progetti di Lucio possono essere lette sul suo blog: http://blog.flyingpic24.com

1) https://www.microchip.com/cip

2) https://www.microchip.com/curiosity

3) https://www.microchip.com/mcc

4) https://github.com/luciodj/rocket

5) This is not Rocket Science – Lucio Di Jasio - Lulu.com - 2015 - ISBN:9781312907775

6) https://github.com/luciodj/simplicity

IL SISTEMA ELETTRICO A 48 VOLT: IL PRIMO PASSO VERSO LA MOBILITÀ ELETTRICA

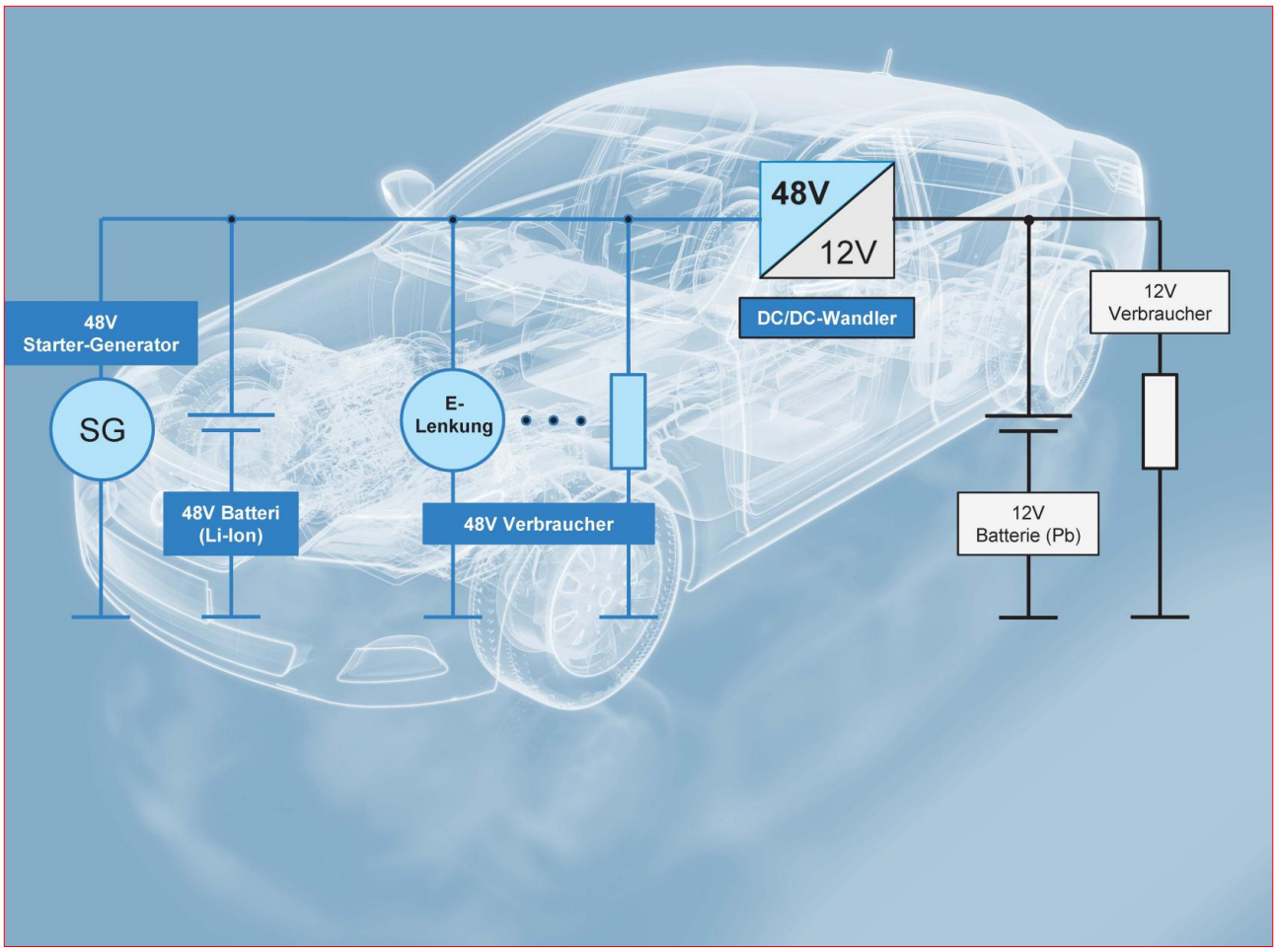
di Andreas Mangler

Per stare al passo con i limiti sempre più severi imposti sulle emissioni di CO2, il passaggio verso il sistema elettrico a 48 Volt è inevitabile. Che cosa significa per i componenti elettronici questa "trasformazione" delle applicazioni convenzionali di potenza nel veicolo caratterizzata da livelli più alti di tensione?

Lo scandalo delle emissioni lo ha messo in luce: gli attuali limiti sulla CO2 sono praticamente irraggiungibili con i motori a combustione interna. Molti Paesi renderanno ancora più restrittivi i propri requisiti negli anni a venire: l'Europa ridurrà l'attuale limite di 130 g/km a 95 g/km nel 2020, mentre gli Stati Uniti sono pronti a seguire nel 2015 con un limite di 102 g/km. Quale fase intermedia verso la mobilità elettrica, i potenti azionamenti ibridi offrono soluzioni ben definite in varie topologie e configurazioni. Da questo punto di vista, le varianti mild hybrid, che incorporano le funzioni di avviamento/motore/generatore, rappresentano un importante passo verso l'elettrificazione dell'avantreno. Uno dei prerequisiti di base è il sistema elettrico a 48V. Questo perché il sistema elettrico a 12 Volt di per sé richiederebbe correnti molto alte. Inoltre, non è più in grado di soddisfare in modo accettabile i requisiti crescenti di un numero sempre maggiore di consumatori. Per continuare ad usare i componenti caratterizzati da bassi consumi di potenza, molti fornitori stanno attualmente mantenendo il sistema elettrico a 12 Volt. Secondo la VDA (l'Associazione tedesca dell'industria automobilistica), nel 2020 circa un milione di veicoli sarà dotata di sistemi elettrici ausiliari a 48 Volt. Entro il 2026, questa cifra crescerà fino a 10 milioni di veicoli, pari a circa un veicolo ogni dieci nel mondo. Ciò richiede un cambiamento totale dell'architettura del veicolo. Essendo la tensione quattro volte superiore, le correnti devono e/o possono essere ridotte al fine di mantenere la potenza invariata. Il requisito principale per assicurare l'integrazione ottimale del motore elettrico all'interno della topologia del veicolo è che possa passare anche alla modalità generatore. Ciò consente il recupero dell'energia di frenata, ossia il processo di recupero dell'energia persa durante la frenata e la sua restituzione alla batteria. Questa operazione è nota come guida a veleggio/motore spento senza emissioni. In tale scenario, la velocità del veicolo in operazioni di marcia libera è mantenuta usando l'energia recuperata dalla batteria. Sono inoltre possibili l'ingresso e l'uscita da un'area di parcheggio e l'accelerata elettrica senza emissioni di CO2. In più, il sistema elettrico ausiliario a 48Volt crea una base per poter introdurre ulteriori innovazioni senza costringere i fornitori a ricorrere a motori più grossi.

Il Mild Hybrid

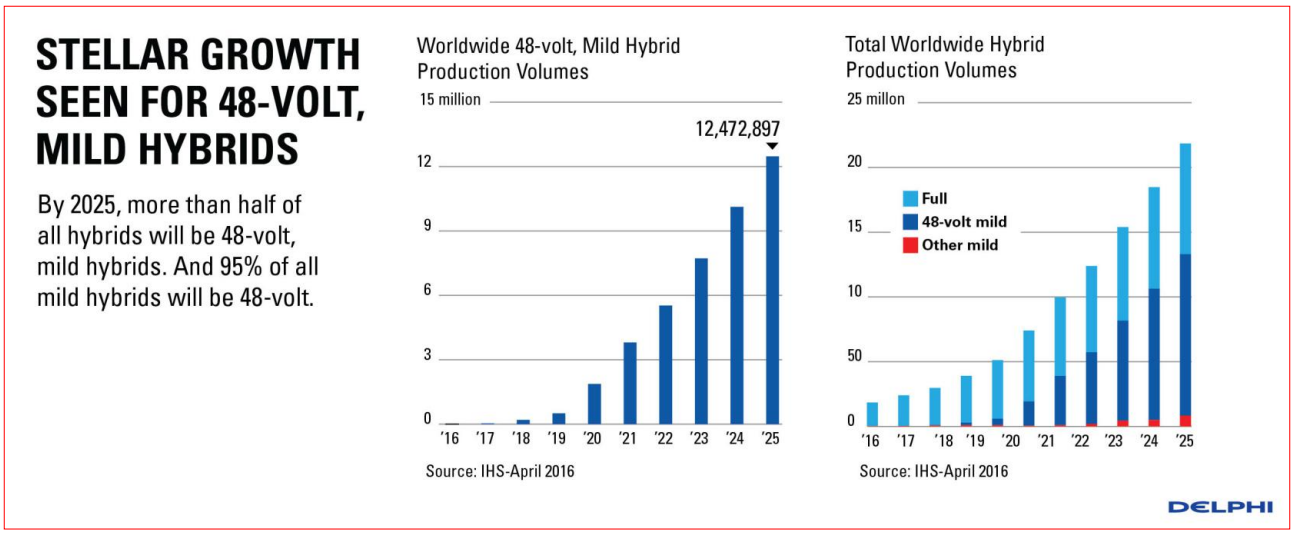
Il sistema elettrico ausiliario a 48 Volt in questi tipi di modelli mild hybrid dovrebbe essere visto come il primo passo verso la tecnologia ibrida e quindi come primo passo nella direzione della "autentica" elettrificazione dei veicoli come



full hybrid. Diversamente dai modelli full hybrid che operano a tensioni significativamente più alte, fino a 360 Volt, i modelli mild hybrid sono configurati senza una funzione di carica. Dato che entrambi gli avantreni non sono completamente disaccoppiati, essi non permettono la guida puramente elettrica, ma semplicemente la guida a motore spento, un sistema start-stop automatico e l'avvio graduale. Pur tuttavia, la guida senza motore da sola riduce le emissioni di CO2 anche del dodici per cento, in relazione alle modalità di misura delle emissioni e allo stile di guida. La decisione di introdurre un sistema elettrico ausiliario a 48 Volt si traduce in specifiche ben precise per l'OEM, per i fornitori e per l'elettronica.

Il Recupero

Durante il recupero, viene ricavata una potenza compresa fra 3kW e 12kW - che per 12kW corrisponderebbe a non meno di 1000 Ampère a 12 Volt! Nel sistema elettrico a 48 Volt, questo valore è ridotto ad un quarto, o al massimo a 250 Ampère. Tuttavia, la tensione più alta richiede una maggiore attenzione verso i requisiti di sicurezza. Per assicurare la conformità alla direttiva Europea sulla bassa tensione, si devono evitare i picchi di tensione oltre i 60 V. Per garantire che il circuito funzioni in sicurezza, è necessario definire l'intervallo operativo ottimale e implementare misure per la sicurezza che offrono protezione contro le sovratensioni e le sottotensioni. La potenza di recupero dipende tra l'altro dalle dimensioni del motore: le perdite per frizione aumentano proporzionalmente con il numero dei cilindri e la potenza di recupero si riduce. Per questo motivo per i motori di piccole dimensioni si consiglia un accoppia-



mento fisso fra il motore meccanico e il generatore di avviamento. Per motori di dimensioni superiori, è più indicata una connessione separata e con disaccoppiamento attivo, perché ciò consente di reimmettere l'energia di frenata nella batteria.

Riduzione del Peso

Grazie alle correnti più basse presenti nel sistema elettrico a 48 Volt, è possibile usare cavi con sezioni trasversali significativamente più piccole. Di conseguenza, il peso del veicolo è ridotto anche di 10 Kg. In più, i componenti meccanici, come lo sterzo, possono essere sostituiti con unità elettriche - più piccole e più leggere.

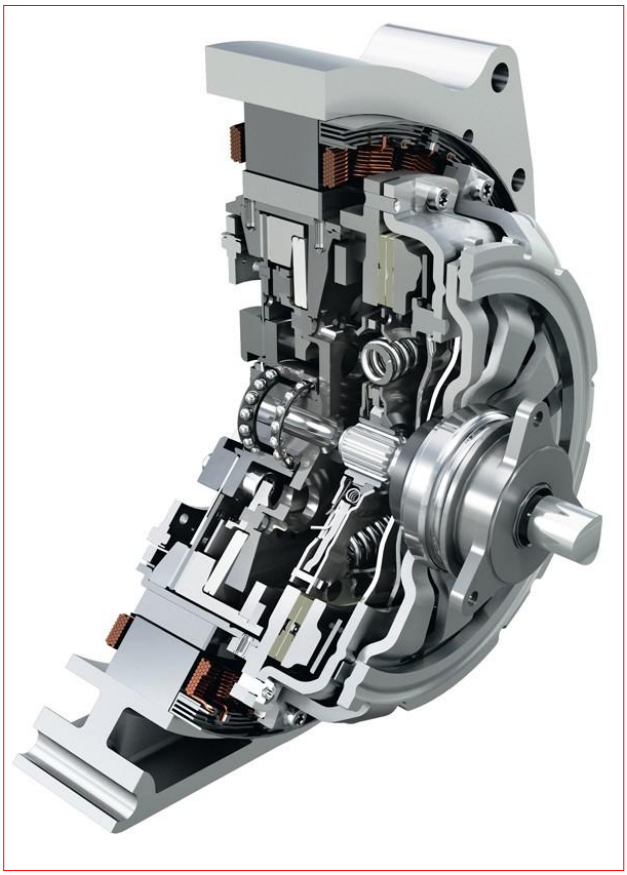
Consumi inferiori, maggiore Comfort

Per l'integrazione all'interno del sistema elettrico ausiliario a 48 Volt, gli elementi più indicati da cui cominciare sono gli assemblaggi caratterizzati da consumi di potenza elevati, come i riscaldatori ausiliari PTC, i sistemi di riscaldamento dei sedili o del lunotto anteriore, oltre al sistema di controllo del clima.
Nel medio termine, a questi ultimi si aggiungeranno altri componenti come lo sterzo elettrico, gli stabilizzatori dello chassis, gli amplificatori audio di potenza e l'illuminazione a LED, che potranno quindi operare a tensioni più alte. Nel lungo termine, essi saranno seguiti da elementi di dimensioni inferiori che consumano potenza. Questo consentirà così di combinare praticità e comfort, dal momento che tali modifiche producono spesso un miglioramento del comfort alla guida. Ad esempio, se il riscaldamento del lunotto anteriore e posteriore è effettuato a una potenza superiore, pari a 1,5kW a 48 V, le ventole possono ruotare ad una velocità inferiore. Nel sistema elettrico a 48 Volt è inoltre disponibile il riscaldatore ausiliario PTC con piena potenza alla velocità minima e attivo immediatamente dopo l'avvio del motore. Esso consente di riscaldare l'abitacolo, il motore e la trasmissione prima che il motore si riscaldi. Ciò è importante in particolare nei veicoli ibridi, dato che il motore elettrico non genera virtualmente calore di dissipazione. Riguardo al sistema di start/stop automatico del motore, il sistema elettrico a 48 Volt consente anche un avvio rapido e lineare senza l'intervento del motore a combustione interna, ad esempio per le operazioni di stop/start nel traffico. Se il compressore per il sistema di controllo del clima è connesso al

sistema elettrico a 48 Volt, il riscaldatore PTC, la ventilazione e le ventole oltre al riscaldatore a contatto possono continuare ad operare ancora per circa tre minuti, anche durante un arresto automatico del motore.

I Componenti nel Sistema Elettrico a 48 Volt

I componenti chiave del sistema elettrico a 48 Volt sono il generatore start/stop, la batteria - basata attualmente sulla tecnologia agli ioni di litio - e inoltre lo sterzo elettrico e i riscaldatori. Tutto ciò richiede un nuovo progetto di tutte le unità di controllo corrispondenti, per assicurare il rispetto dei nuovi requisiti che si applicano in presenza di tensioni più alte. Il generatore di avviamento e l'elettronica di potenza possono essere realizzati con una macchina di tipo claw-pole o con una macchina sincrona. La variante claw pole è più strettamente allineata con lo standard a 12 Volt, permette la produzione in massa a costi convenienti ed è più semplice da integrare all'interno dei nuovi veicoli. Tuttavia, la cinghia trapezoidale ha un'influenza negativa sulla trasmissione della coppia, e la durata utile del sistema ad anelli collettori è limitata. Nella macchina sincrona il magnete permanente assicura la massima densità di potenza. Non usando gli anelli connettori, è praticamente esente da manutenzione. Gli svantaggi sono i costi più elevati, l'integrazione più complessa e l'impiego di materiali ricavati da terre rare per la sua realizzazione. La batteria include le celle stesse della batteria, un sistema di raffreddamento, l'alloggiamento specifico e la struttura a pila, il sistema di bilanciamento delle celle e un sistema sofisticato per la gestione delle batterie. Quest'ultimo si avvale di varie funzioni diagnostiche, come il monitoraggio della temperatura, il controllo delle sovratensioni e delle sottotensioni, per assicurare che le celle della batteria rimangano all'interno del proprio intervallo operativo di sicurezza. Sotto questo aspetto, occorre considerare un'intera gamma di obiettivi di progetto, in particolare la carica rapida a 12 kW, l'alta densità di energia, la conformità ai requisiti anticollisione e inoltre il raggiungimento di una lunga durata utile accompagnata alla minimizzazione dei costi. Le batterie agli ioni di litio soddisfano gran parte di questi requisiti. Le celle cilindriche offrono anche il vantaggio di essere disponibili in dimensioni standard ed economicamente convenienti e di poter essere connesse in parallelo e in serie in modo da formare una pila. La determinazione dello stato di salute (S-o-H) delle celle della batteria impone un'ulteriore sfida. Gli attuali siste-



mi di gestione delle batterie aderiscono a procedure di documentazione stringenti per registrare le operazioni di carica e di scarica. Durante questo processo, essi accumulano vaste quantità di dati dovuti delle complicate procedure di conteggio delle cariche, essenzialmente per trarre conclusioni in merito allo stato della batteria. La misura delle impedenze complesse in condizioni reali come mezzo per determinare lo stato di salute della batteria non è ad oggi né economicamente conveniente né pronta per la produzione in serie. È precisamente in questo ambito dei sistemi di gestione delle batterie che Rutronik sta attivamente promuovendo la ricerca, in collaborazione con le principali università. Un componente chiave del sistema elettrico ausiliario a 48 Volt è anche il convertitore DC/DC bidirezionale che connette entrambi i sistemi. Questa unità offre tipicamente un'uscita a 3kW con picchi anche di 3,5 kW per 2 secondi ed è raffreddata passivamente per ottenere un'efficienza di almeno il 96 per cento. Diversi fornitori stanno già offrendo soluzioni che soddisfano i nuovi requisiti del sistema elettrico a 48 Volt. Ad esempio, il fornitore leader Valeo ha sviluppato delle soluzioni a supporto dell'avantreno. Altri, come Continental, Bosch, Preh o Delphi stanno sviluppando indipendentemente delle soluzioni. Tuttavia, dato che né i veicoli mild hybrid, né i veicoli completamente ibridi saranno visti in massa sulle strade nell'imminente futuro, in termini finanziari il segmento è relativamente di scarso interesse per il settore della distribuzione - per il tempo a venire. Ma quando si tratterà di rispettare le normative sulla CO2, non ci sarà alternativa. Un ulteriore incentivo è fornito dai decisori politici, con il loro programma di supporto alla mobilità elettrica. L'obiettivo è di immettere mezzo milione di veicoli elettrici sulle strade entro il 2020, con l'aiuto dei finanziamenti destinati ai veicoli elettrici, i quali includono anche i modelli ibridi, con gli incentivi fiscali e con l'introduzione delle stazioni di ricarica. Per raggiungere questo obiettivo, la Divisione Automotive di Rutronik offre ai propri clienti una gamma di componenti scelti con cura da fornitori selezionati, oltre ad un supporto tecnico e commerciale specifico a livello globale. In stretta cooperazione e coordinamento con i fornitori selezionati e con i loro componenti di ultima generazione specifici per applicazioni automotive, la Divisione Automotive supporta i clienti chiave nell'affrontare le attuali sfide per un'ampia gamma di applicazioni automotive, in particolare nell'area della e-mobility. Altre aree di interesse per queste attività ed applicazioni includono la comunicazione fra veicoli, l'infotainment, i sistemi di guida assistita e l'illuminazione, oltre ai requisiti di sicurezza in continua crescita basati sullo standard ISO 26262, il quale copre anche le applicazioni elettriche a bordo del veicolo.

Il Supporto ai Clienti garantito a Livello Mondiale

Con la propria vocazione internazionale basata sulla cooperazione, la Divisione Automotive offre ai clienti il supporto ottimale in tutto il mondo. Per il supporto locale, essa opera con gli ingegneri applicativi locali di Rutronik e con i responsabili per lo sviluppo aziendale.

Registratore consumi elettrici (prima puntata)

di Grazia Ancona

La forte attenzione al risparmio energetico è mossa principalmente da due componenti: il rispetto ambientale ed un'attenta gestione delle risorse economiche. In questo articolo vi mostriamo come interpretare e valutare i consumi domestici o aziendali rilevando il loro andamento nell'arco di 12 o 24 ore. La curva di consumo reale così ottenuta ci aiuterà a correggere eventuali sprechi d'energia cogliendo, in tal modo, i benefici a cui abbiamo accennato.

La potenza istantanea e la potenza utilizzata

Tutti noi conosciamo il concetto di potenza elettrica: essa è il prodotto della tensione ai capi di un utilizzatore moltiplicata per la corrente che l'attraversa. Questo concetto, valido per le correnti continue, è quasi (e sì, c'è sempre un "quasi" di mezzo…) perfettamente assimilabile anche per le correnti alternate periodiche sinusoidali fornite dai vari gestori. Naturalmente, in questa breve trattazione, ci occuperemo di queste ultime. La potenza elettrica è misurata in W (Watt) ma per scopi energetici domestici si preferisce utilizzare il KW, equivalente a 1000 W. Riassumendo, possiamo dire che ai capi di un carico puramente resistivo (ad esempio la resistenza di un boiler o una stufa elettrica) alimentato a 220 V alternati, la potenza elettrica che sarà convertita in calore (fatto salvo alcune perdite di cui per semplicità non ci occuperemo) è pari a:

$$W = 220 \times I$$

In cui I è la corrente assorbita dal carico. Ad esempio se questa corrente è pari a 4 A alternati avremo una potenza pari a 220x4=880 W (0,88 KW). Diremo, a proposito dell'esempio, che il valore così calcolato è la

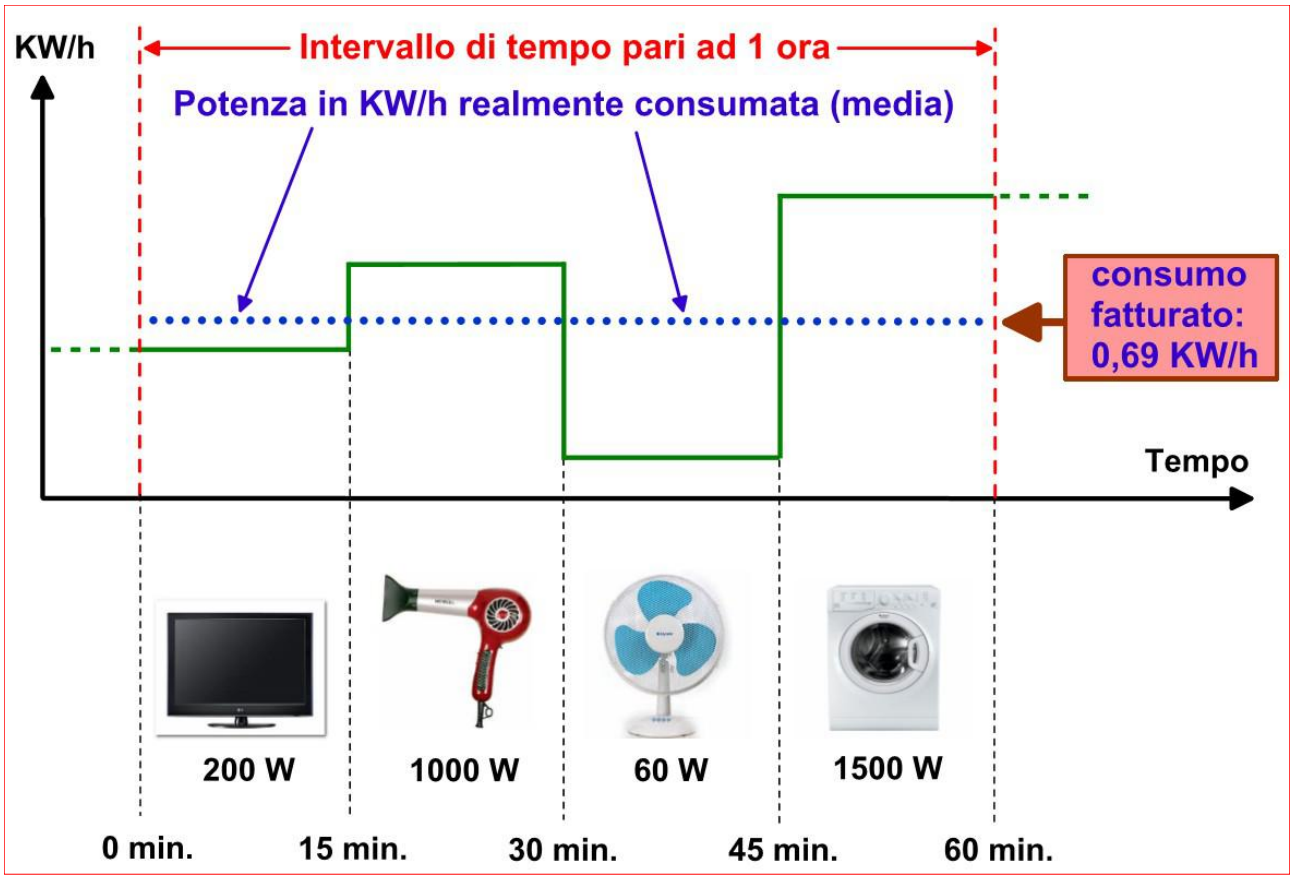


Figura 1: Totalizzazione media dei consumi elettrici in un'ora.

potenza istantanea. Infatti nessuno può prevedere se, un attimo dopo, l'utente spegne il suo asciugacapelli per ricaricare il telefono cellulare. Inoltre, se il carico non è perfettamente resistivo ma presenta componenti induttive (ad esempio una lavatrice) o capacitive (ad esempio alcuni vecchi alimentatori per PC) il calcolo sopra visto si complica un pochino. Tuttavia, in ogni caso, utilizzeremo dell'energia che ci viene fornita da gestore esterno e che sarà fatturata proporzionalmente al suo consumo. La tensione alternata della rete elettrica è comunque affetta da una certa tolleranza, in particolare può variare del più o del meno fino ad un massimo del 10%. Ovvero, ai capi delle nostre prese, potremmo trovare 198 VCA in un momento di massimo "calo" così come, ad esempio durante la notte, possiamo misurare fino a 242 VCA. A fronte di ciò, il gestore preferisce calcolare la potenza da noi utilizzata direttamente in KW. Nel compiere questa operazione non farà una misura istantanea e non utilizzerà una media di più misure rilevate. Semplicemente calcolerà il consumo continuo in un'ora. Cosicché il nostro KW ora diventa KWh (Kilowattora) e rappresenta la totalizzazione (integrazione nel tempo) di tutti i consumi elettrici dell'utente nell'arco di un'ora (vedi figura 1). Tutto questo pur considerando la fluttuazione della tensione alternata dovuta alla sua tolleranza e l'inserimento, nella linea elettrica, di carichi non esattamente resistivi.

Il contatore fornito dal gestore energetico

La misura dell'energia utilizzata è fatta da un apposito strumento che, familiarmente, è chiamato "contatore". In realtà il suo vero nome dovrebbe essere "wattometro orario" ma noi, per semplicità, continueremo a chiamarlo con il suo nome al secolo. Tutti noi abbiamo ben presente il vecchio contatore a disco mobile, ormai non più in uso, sostituito da nuovi contatori elettronici gestibili direttamente dalla centrale di distribuzione. L'utilizzo di questi moderni strumenti di lettura ha colto i seguenti vantaggi:

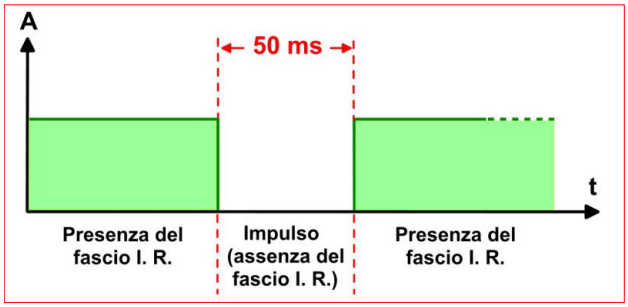


Figura 2: Impulso proveniente dal LED I.R.

- Assoluta impossibilità di contraffazione da parte dell'utente circa i reali consumi.
- Gestibilità completa dalla centrale dell'ente erogante con conseguente riduzione dei costi d'intervento del personale (telegestione).
- Maggiore precisione nella rilevazione dei consumi di potenza attiva.
- Indicazione quantitativa della potenza utilizzata da carichi a forte componente induttiva o capacitiva (potenza reattiva, misurata in KWARh). Questo tipo di carico non è permesso dal contratto se supera una certa soglia.
- Disponibilità, anche da parte dell'utente, di informazioni di carattere tecnico e fiscale (alcune sono presentate su un display LCD).

Il moderno contatore ha anche un'altra importante funzione che, in precedenza, era svolta da dispositivi ausiliari, sempre forniti dal gestore. Stiamo accennando all' interruzione della linea interna dell'utente in caso di sovraccarico. A questo proposito c'è da dire che il gestore, su un impianto standard per casa privata da 3 KW massimi, accetta un sovraccarico del 10% in più. Ovvero potremmo consumare fino a 3,3 KW senza che vi sia interruzione di corrente dovuta agli organi disgiuntori, quest'ultimi integrati nel contatore stesso. L'interruzione è ovviamente garantita anche in caso di corto-circuito. Tuttavia, in questa eventualità, dovrebbe aprirsi anche l'interruttore differenziale magneto-termico che il gestore ci impone di installare, a nostre spese e responsabilità, subito a valle del contatore. Quello che il contatore per adesso non è in grado di fare è mostrarci una relazione precisa e leggibile di quelli che sono i nostri consumi nell'arco di 24 ore. Ed è proprio questo, invece, il compito della nostra applicazione: informarci, con un grafico accurato, dei consumi elettrici per ogni momento della

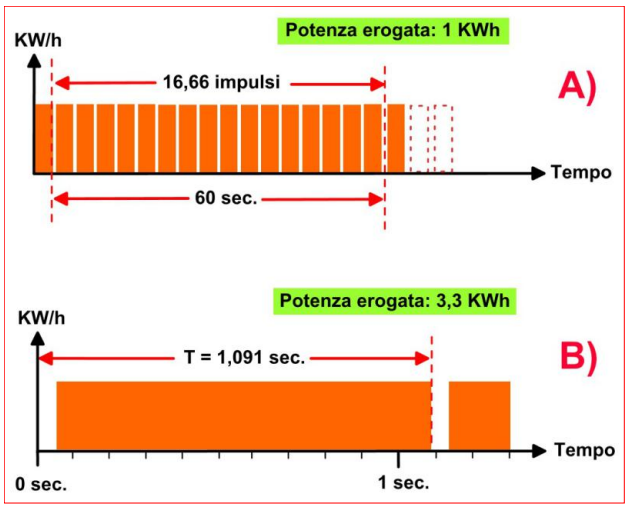


Figura 3 A-B: Impulsi contati in un minuto, in un secondo e potenza erogata corrispondente.

giornata. I dati acquisiti ci aiuteranno a razionalizzare i consumi, magari rivedendo la fascia d'utilizzo di elettrodomestici particolarmente "energivori", C'è da dire, inoltre, che alcuni gestori applicano sconti a consumi prodotti in fasce orarie di riposo energetico (la notte o il primo mattino). Anche in questi casi, la nostra applicazione, potrà rivelarsi di notevole aiuto.

Il principio di misura della potenza erogata a mezzo LED infrarosso

Come accennato, i moderni contatori hanno la possibilità di visualizzare informazioni di vario tipo, a mezzo visualizzazione su display LCD, anche dati relativi al consumo totalizzato dall'utente. Ovviamente, prendendo nota dei dati del giorno prima e confrontandoli con l'esatta ora del giorno dopo, si avrà il consumo totale giornaliero. Tuttavia questo dato, pure importante, non ci dice nulla di come è distribuito il consumo nell'arco della giornata presa in considerazione. I progettisti del contatore hanno però considerato la possibilità di esaminare in tempo reale la potenza erogata, lo hanno fatto in due diversi modi:

1. Utilizzando un apposito simbolo (un puntino di medie dimensioni) sul display LCD.
2. Utilizzando un diodo LED ad emissione infrarossa posto sul pannello frontale del contatore.

La filosofia di funzionamento, quasi identica in entrambi i casi, è molto semplice: il consumo è proporzionale al numero di lampeggi nell'unità di tempo. Il simbolo grafico su LCD (tra l'altro non presente in tutte le versioni di contatore) ha una mera funzione indicatrice. Ovvero, a meno di non avere molto tempo a disposizione ed un buon cronometro, non può esserci d'aiuto nella nostra applicazione. Utilizzeremo invece il LED I.R. che, nei manuali forniti a corredo del contatore, è indicato con la dicitura "SEGNALE OTTICO KWh AD USO TECNICO". Naturalmente il segnale emesso. essendo nello spettro infrarosso, non è visibile da occhio umano. La modalità con cui questo LED ci informa del consumo sono semplici: un fascio I.R. continuo è interrotto da impulsi aventi fronte iniziale negativo e forma rettangolare (vedi figura 2). La scelta di utilizzare impulsi a fronte negativo, apparentemente poco ovvia, è invece orientata a garantire la rivelazione continua del fascio I:R. Infatti, qualsivoglia mancanza della portante infrarossa, magari dovuta ad un mancato allineamento della sonda ottica, può essere avvertita dal dispositivo di lettura in modo semplice e diretto. Viceversa, disponendo di impulsi a fronte iniziale positivo, la mancanza del fascio I.R. poteva confondersi con il mancato prelievo di energia da parte dell'utente (zero consumi). Il protocollo di modulazione del fascio I.R., come accennavamo, è semplice: il numero degli impulsi nell'arco di un'ora esprime la potenza utilizzata in KWh. In particolare viene osservata la relazione:

1 KWh = 1.000 impulsi x ora

Vedremo meglio, più avanti, eventuali varianti quantitative di questa formula, derivanti dal ti-

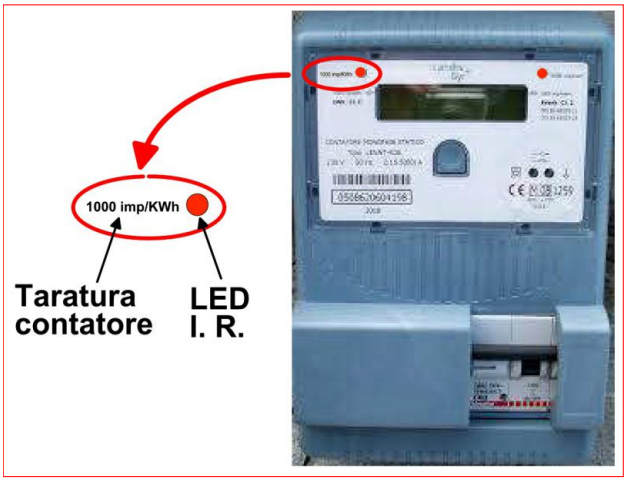


Figura 4: Taratura in impulsi per KWh sul contatore.

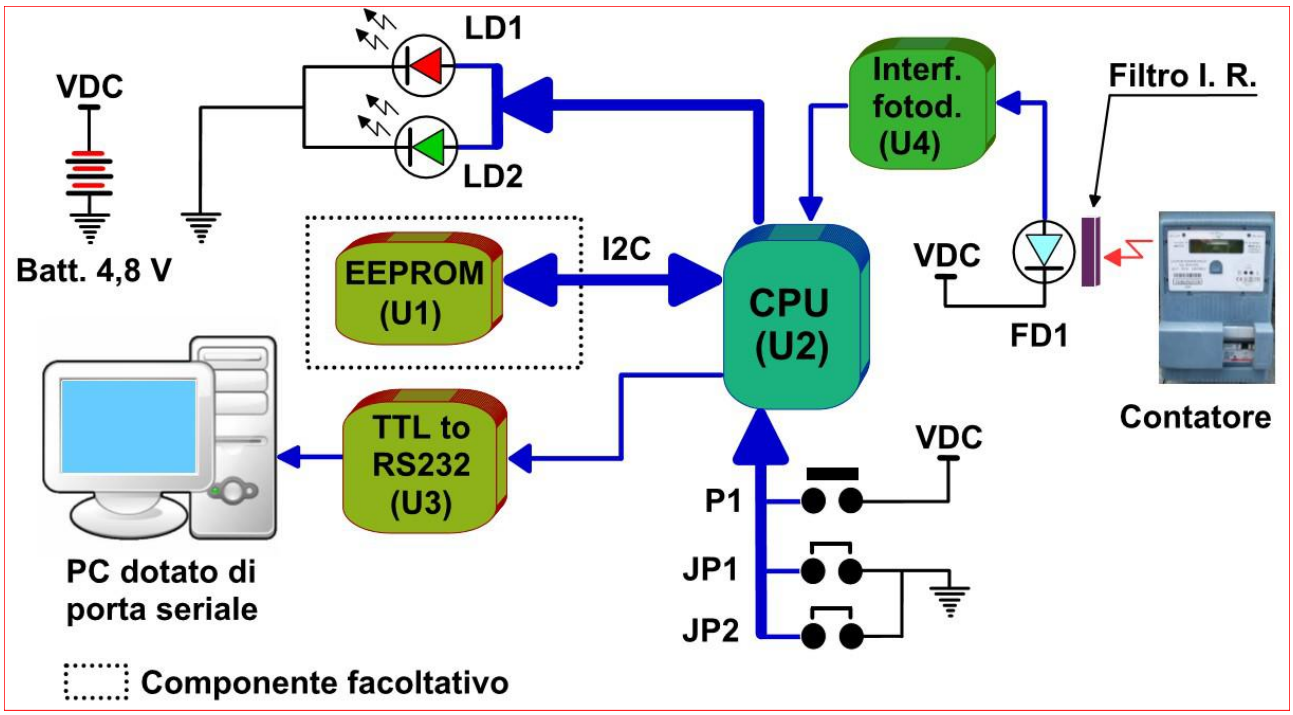


Figura 5: Schema a blocchi dell'applicazione.

po di contatore fornito. Tuttavia il concetto è sempre lo stesso: contare il numero degli impulsi nell'unità di tempo per avere una misura della potenza erogata. Ovviamente, nota la relazione, possiamo anche decidere di analizzare il numero degli impulsi in un arco di tempo più breve, ad esempio un minuto (vedi figura 3A). Infatti, sapendo che 1.000 impulsi in un'ora sono equivalenti ad 1 KWh, è facile dedurre quanti impulsi sono necessari, nell'arco di un solo minuto per una potenza equivalente. Sapendo che un'ora è composta da 60 minuti si otterrà:

1 KWh = (1.000 / 60) imp. x 1 minuto =
= 16,66 imp. ogni minuto

E' anche possibile una misura per secondo, in questa guisa il dato è sicuramente più facilmente elaborabile per i nostri scopi. Tenendo conto che un'ora è formata da 3600 secondi si otterrà:

1 KWh = (1.000 / 3600) imp. x secondo =
= 0,27 imp./sec

Supponiamo adesso di utilizzare la massima energia disponibile per un impianto da 3 KW ovvero 3,3 KWh. In questo caso il calcolo deve tener presente anche del rapporto di potenza, ovvero:

3,3 KWh =
= ((3300/1000)x1000)/3600) imp. x sec. =
= 0,91 imp./sec

Quando la frequenza degli impulsi è inferiore all'unità è più conveniente pensare in termini di periodo (vedi figura 3B). In questo caso il periodo di ripetizione degli impulsi sarà pari al reciproco della loro frequenza al secondo:

Periodo di ripetizione impulsi a 3,3 KWh =
= 1 / 0,91 = 1,098 secondi

Una volta assimilato il concetto sarà molto facile progettare un'applicazione che, contando gli impulsi in una certa unità di tempo, ci fornirà dei dati di consumo ad intervalli regolari. Questi dati, raccolti in un grafico, ci daranno un'idea esatta del prelievo d'energia nell'arco di una giornata. C'è da dire che nelle nostre formule abbiamo sempre considerato un conteggio per singolo KWh pari a 1000 impulsi/ora. Questo non è sempre vero poiché esistono contatori che, nell'ambito di un'ora, forniscono invece 10.000 impulsi per un KWh. Un maggior numero di impulsi orari consente una migliore precisione della lettura, tuttavia è stato necessario tenerne conto durante lo sviluppo dell'applicazione. Infatti, non

possiamo sapere a priori che tipo di contatore ha l'utente che sta utilizzando il nostro registratore di consumi. Il problema è stato risolto nel firmware che, a sua volta, ne è informato per mezzo di un DIP-FIX che sarà predisposto dall'utente. Quest'ultimo, per un'esatta impostazione, dovrà leggere i dati presenti alla sinistra del LED I. R. del proprio contatore: Il tutto è riassunto graficamente in figura 4.

Le specifiche della nostra applicazione

In base a quanto esposto, sarà ora possibile progettare un hardware che ci permetta di soddisfare tutte le considerazioni fatte nella nostra breve analisi. Tuttavia occorre ancora qualche ulteriore considerazione. Intanto il nostro dispositivo deve essere semplice ed economico poiché, molto spesso, il contatore può trovarsi in locali comuni, facilmente accessibili, oppure all'aperto. In questi casi, essendo sottoposto alla "pubblica fiducia", è bene valutarne la perdita per asportazione. Ne consegue che il valore materiale dell'applicazione deve essere, abbastanza modesto. Niente display o componenti costosi, l'interfaccia con l'utente deve essere più semplice possibile: un semplice pulsante. La visualizzazione dei stati del dispositivo è affidata a due semplici LED e la sonda I. R. è un comunissimo ed economicissimo fotodiodo che sarà, vedremo meglio in seguito come, opportunamente filtrato. L'alimentazione dovrà necessariamente essere costituita da batterie ricaricabili in quanto, sembra un controsenso, nei pressi del contatore potrebbero non esserci prese elettriche. Tale necessità implica un'attenzione particolare ai consumi ed una meticolosa scelta delle soluzioni progettuali. Abbiamo accennato al fatto che il prodotto di questo registratore di consumi è un grafico su foglio EXCEL. Di conseguenza occorrerà un interfaccia in grado di inviare i dati direttamente sulla seriale del PC. Avendo nella nostra scheda pre-cablata (già interessata ad altre applicazioni) un'interfaccia di tipo RS232, abbiamo pensato di utilizzarla per diminuire i tempi di sviluppo. Il lettore, se lo desidera, potrà usare un altro tipo di protocollo (ad esempio usando la porta USB) modificando leggermente lo schema elettrico e una piccola parte del firmware. Purtroppo, il programma EXCEL non è in grado di accettare direttamente i dati in caratteri ASCII provenienti dalla porta seriale. Per ovviare a questa limitazione,, è stato implementato come macro EXCEL un piccolo programma in BASIC che permette ai dati l'accesso al livello di presentazione-visualizzazione. Per una maggiore comodità di utilizzo, la nostra scheda è dotata di una memoria EEPROM (opzionale) in grado di "ricordare" i dati raccolti. E questo anche se nel frattempo il dispositivo è stato spento o

Tabella 1: Elenco componenti

B1	4 batterie ricar. 1,2 V 1-1,2 Ah
C1,C12	100 nF 50 VL poliest
C2,C3	10 uF 16VL elettrol
C4,C7,C8, C9,C10 C5,C6	2.2 uF 25 VL elettrol
C15,C16	22 pF ceramico disco
C14	1 nF 50 VL ceram disc
D1	1N4148 Diodo G. P.
FD1	BPW34 Fotodiodo IR
J1	Conn. DB9 maschio
J2	Morsett vite 5 cont
JP1,JP2	Dip fix a 2 cont
LD1	Led 3 mm rosso
LD2	Led 3 mm verde
P1	Pulsante NA
R1,R2	10 K Resist 1/4 W 2%
R11,R12	22 K Resist 1/4 W 2%
R6,R7	560 Resist 1/4 W 2%
R8	68 K Resist 1/4 W 2%
R3,R4,R5	4,7 K Resist 1/4 W 2%
R9,R10	100 K Resist 1/4 W 2%
U1	24C08L EEPROM 1Kbit I2C Bus (DIP 8)
U2	PIC18F252 Microcontr (DIP 28)
U3	MAX 232 Convertitore TTL RS232 (DIP 16)
U4	BC558 Transistor PNP (TO92)
X1	4,00 MHz Quarzo subminiatura
X2	33768 Hz Quarzo subminiatura

Ordine delle fasi	Durata fase	Stato macchina	Operazione utente
1	non determ.	Registratore spento.	Predisposizioni su DIP- FIX (tempo di raccolta dati e impulsi al KWh).
2	2 sec.	Accensione	Osservare i LED che possono evidenziare eventuali stati di allarme.
3	non determ.	Attesa del comando utente.	Avviare la fase di registrazione oppure (se possibile) dump dei dati raccolti.
4	12 h o 24 h	Raccolta dati dal contatore secondo il tempo impostato.	Attesa della fine della fase di campionamento (12 o 24 ore)
5	non determ.	Attesa del comando utente.	Avviare la fase di dumping oppure iniziare un nuovo campionamento.
6	2 sec.	Versamento dei dati su porta seriale	Visione dei dati raccolti su foglio EXCEL.

Figura 6: Tutte le fasi richieste per ottenere il grafico dei consumi giornalieri.

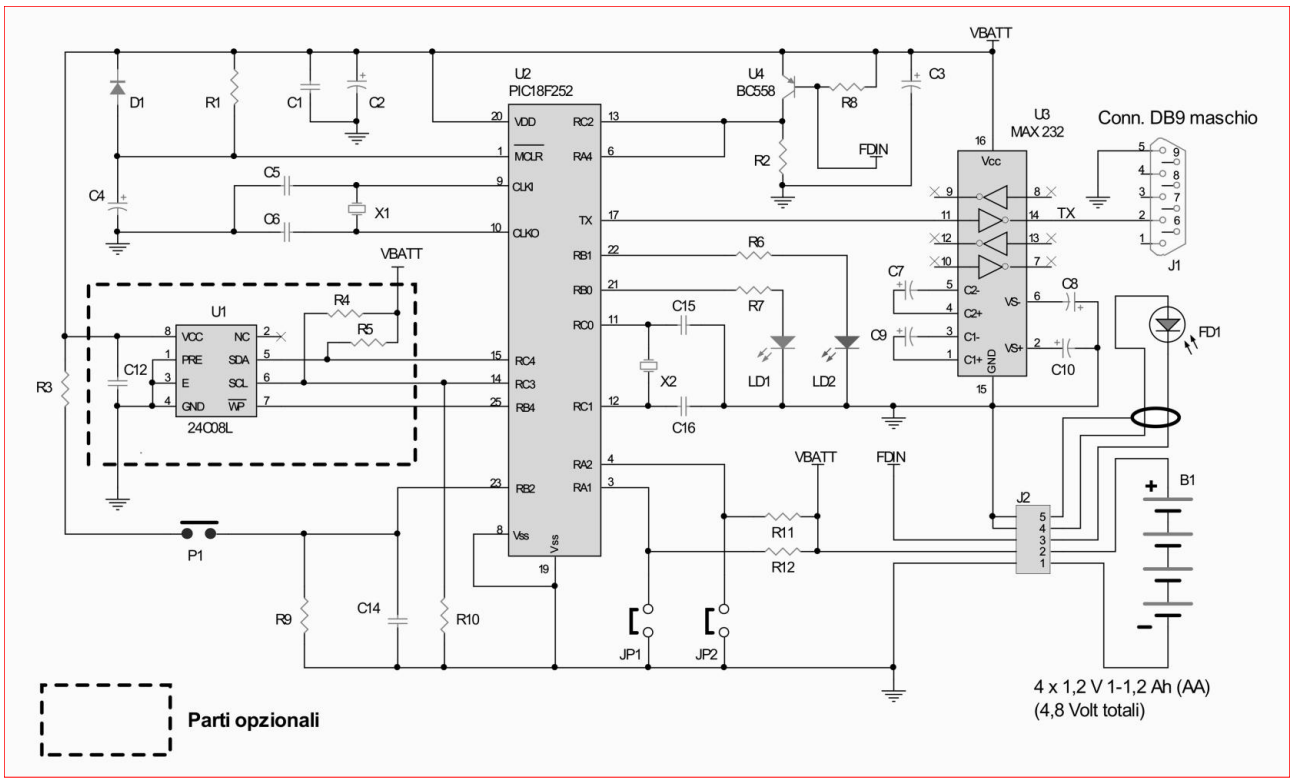


Figura 7: Schema elettrico.

le batterie si sono scaricate del tutto. In figura 5 possiamo vedere la rappresentazione a blocchi di quanto esposto. Come potete notare, il tutto è improntato alla massima semplicità e minimo uso di componentistica. Quindi, riassumendo, la nostra applicazione dovrà essere in grado di raccogliere i dati di consumo energetico in 12 o 24 ore, mantenerli (nel caso sia installata la EEPROM) e poi riversarli su un PC in cui vi sia installato EXCEL. Se l'utente non installa la EEPROM tutte queste operazioni dovranno essere fatte con la scheda che riamane alimentata altrimenti i dati raccolti andranno persi. La figura 6 illustra, a scanso di equivoci, tutti questi passaggi. Di seguito forniamo invece le caratteristiche principali:

- Potenza massima registrabile: 3,5 KW/h.
- Durata campionamento: 12 o 24 ore impostabili dall'utente.
- Cadenza impulsi accettata: 1000 imp./KWh oppure 10000 imp./KWh selezionabili dall'utente in base al contatore presente nell'impianto.
- Risoluzione grafico (su modello fornito): 100 W/h (sono possibili risoluzioni maggiori modificando il foglio EXCEL).
- Errore medio sui dati campionati: inferiore a ± 2% (a 20 gradi centigradi).
- Consumo in fase di campionamento: 8,5 mA.
- Alimentazione: 4 batt. NiCd da 1,2 V con 0,8 A/h (minimo).
- Interfaccia: tipo RS232C (nessun protocollo richiesto).
- Test automatico all'accensione delle principali funzioni hardware.
- Possibilità di attivare un test che monitorizza gli impulsi provenienti dal contatore.
- Comandi a disposizione dell'utente: singolo pulsante.
- Visualizzazione stati macchina: a mezzo due LED (rosso e verde).
- Possibilità di mantenere i dati raccolti per un tempo lunghissimo anche con il dispositivo spento.
- Temperatura d'utilizzo: da 0 a 50 gradi centigradi.

Analisi dello schema elettrico

Tutto questo bel dire si riassume nello schema elettrico di figura 7 e nella lista componenti di tabella 1. Come premesso, si conferma la grande semplicità ed economia dell'insieme. Le interfacce utente sono ridotte

Funzione dei pin del PIC 18F252				
PIN		USO	DESCRIZIONE FUNZIONE	NOTE
4	RA2	in dig.	ingresso predisposizione tempo di campionamento: 12/24 h	Pull-down su GND (L = 12 h) abbinato a JP2.
3	RA1	in dig.	ingresso predisposizione conteggio: 10000 imp./Kw o 1000 imp./Kw.	Pull-down su GND (L = 10000 imp/Kw) abbinato a JP1.
23	RB2	in dig.	Pulsante P1 con funzioni varie (vedi articolo).	Pull-down su GND (H = attivazione).
21	RB0	out dig.	LED rosso con funzioni varie (vedi articolo) acceso se = H	Monitor degli impulsi provenienti dal contatore in fase di TEST, indicatore dello stato della memoria, monitor porta RS232, allarme (in sincronia con LED verde).
22	RB1	out dig.	LED verde con funzioni varie (vedi articolo) acceso se = H	Indicatore di attesa pressione pulsante, indicatore della fase di campionamento, allarme (in sincronia con LED rosso).
27	RB6	PGC	Usati solo in fase di debug e programmazione.	
28	RB7	PGD		
17	RC6	USART	Uscita RS232 (USART TX)	Il segnale è poi convertito in RS232.
25	RB4	out. dig.	WP (bus I2C)	Supportano il protocollo I2C . Se la EEPROM non è presente il pin SCL sarà letto dal processore come "L".
15	RC4	I2C	SDA (bus I2C)	Supportano il protocollo I2C . Se la EEPROM non è presente il pin SCL sarà letto dal processore come "L".
14	RC3	I2C	SCL (bus I2C)	
13	RC2	in dig.	Ingresso impulsi per monitor LED.	Il segnale d'ingresso è fornito da U4 che riceve gli impulsi dal fotodiodo. L'impulso si presenta ai pins del processore con polarità invertita rispetto all'ingresso (presenza impulso = "H").
6	RA4	in dig.	Ingresso contatore TMR0	
11	RC0	CLK out	Pin per quarzo esterno a 32,868 KHh	Clock esterno per timer TMR1
12	RC1	CLK in		
9	-	CLK in	Pin per quarzo esterno a 4 MHz.	Clock di sistema
10	RA6	CLK out		
1	MCLR		Reset RC e uso del VPP	La funzione di VPP è usata solo in fase di debug e programmazione
20	Vdd		Alimentazione positiva	4,8 VDC
8 - 19	Vss		Alimentazione negativa	GND

Tabella 2: Funzione dei pin del PIC 18F252.

al minimo: due LED ed un solo pulsante, quest'ultimo avente tripla funzione. La scelta della CPU, un PIC 18F252 famiglia HIGH PERFORMANCE, potrebbe sembrare affetta da una certa ridondanza di caratteristiche. In effetti, molte possibilità offerte da questa versatile CPU non sono sfruttate. Tuttavia, l'economia del componente, la buona dimensione della memoria programma, la presenza di un interfaccia I2C implementata nell'H.W. e la facile reperibilità (è anche in vendita su I.E: SHOP, come parte di un conveniente bundle, nella sua versione finestrata, comunque usabile in questa applicazione) ne "impongono" la scelta. In tabella 2 possiamo vedere il riassunto dei vari pin secondo la loro funzione nel nostro schema elettrico. Il lettore attento noterà la presenza di 2 quarzi. Dando per quasi scontata la presenza di uno, destinato al clock di sistema, l'altro quarzo rappresenta l'unica particolarità di questo circuito. In effetti il quarzo a 4,00 MHz è il clock generale mentre il quarzo a 32768 Hz è un clock ausiliario i cui impulsi sono posti all'ingresso del contatore TMR1, uno dei 3 contatori di cui dispone il PIC 18F252. In figura 8 possiamo vedere gli elementi interni di TMR1 usati nella nostra applicazione e la loro gestione nel firmware (per semplicità sono stati omessi alcuni blocchi funzionali). Il contatore TMR1 ha un compito importante: è l'orologio che decide i tempi d'intervallo dei campionamenti. Essendo questi pari a 6 minuti, un tempo abbastanza lungo, si è dovuto ricorrere ad un clock di frequenza diversa dal clock di sistema, sensibilmente più bassa: 32,768 KHz. La scelta di questa frequenza, un valore apparentemente insolito, è imposta dalla sua divisibilità con potenze di 2 (16, 32, 64, ecc.) e quindi molto gestibile da parte di divisori binari. Quarzi di questo tipo, oltre ad essere molto economici, sono anche reperibilissimi sul mercato. Tuttavia, anche la scelta di un clock a periodo abbastanza alto non consentiva di raggiungere i fatidici 360 secondi che intercorrono fra un campione ed il

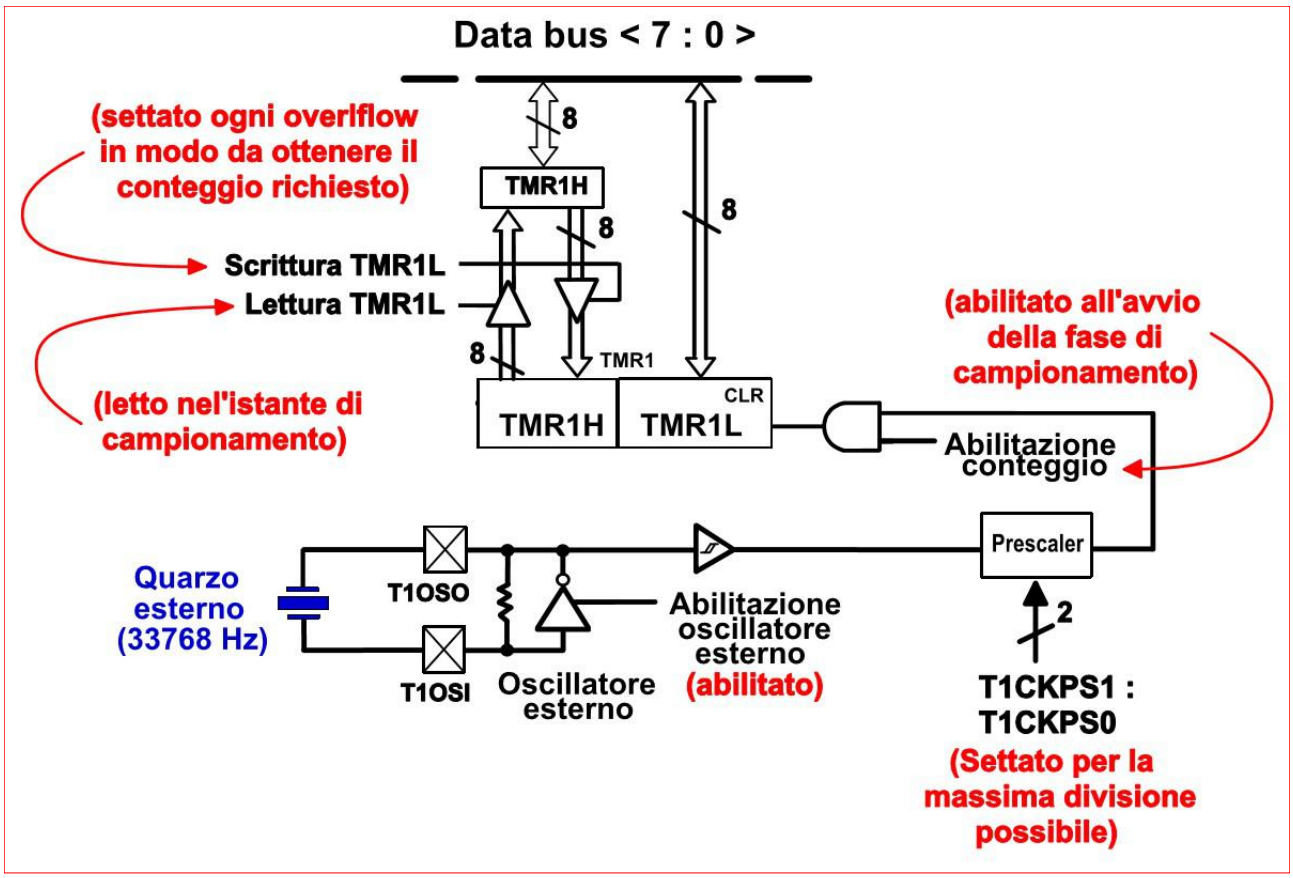


Figura 8: Sezione relativa a TMR1 su PIC 18F252 (semplificata).

successivo. E questo nonostante il prescaler di TMR1 fosse settato al massimo divisore possibile e l'impostazione iniziale dei registri TMR1H e TMR1L (vedi sempre figura 8) utilizzati entrambi per un totale di 16 bit, fosse bassissima. Infatti esiste una relazione per calcolare, dato un input a frequenza costante e nota, il tempo totale di conteggio. Quest'ultimo è inteso come il massimo numero raggiungibile prima che i registri di conteggio, arrivati al loro massimo (ovvero il numero binario 11111111 – 11111111), ricomincino da capo e venga settando il flag di overflow (chiamato TMR1F). La relazione è la seguente:

T = (65536 – impost.contat) x (1 / (32768 / presc)

In cui il valore 32768 è, per l'appunto, il clock del contatore. Dalla relazione risulta, impostando un periodo di 15 secondi e settando il prescaler alla sua massima divisione:

impost.contat = 65536 – (15 / (1 / (32768 / 8)))

Con un tempo massimo di conteggio pari a 15 secondi, l'impostazione iniziale dei contatori sarà il numero 4060. Tuttavia, come accennato, 15 secondi sono ancora troppo poco per i nostri scopi, di conseguenza occorrerà un piccolo aiuto, ovvero un divisore software. Questi non è altro che una variabile ad 8 bit che conta il numero di overflow di TMR1. Essi sono comunicati al programma con un'interruzione che è del tutto "asincrona" dal flusso principale. Sapendo che abbiamo bisogno di 360 secondi, possiamo dedurre facilmente l'overflow di questo ulteriore prescaler software:

Overflow prescaler software = 360 / 15 = 24

Quindi, ogni volta che la variabile del prescaler software è giunta al numero 24, ci sarà un campionamento perché sono passati 360 secondi (ovvero 6 minuti). Questo meccanismo, per funzionare, ha poi bisogno di un'ulteriore variabile che conti il numero dei campioni rilevati. Ricordiamo che essi sono 240 con settaggio a 24 ore e 120 con settaggio a 12 ore. La memoria EEPROM, osservando lo schema elettrico, è posta in un riquadro

Modifiche e predisposizioni da effettuare su modulo EEPROM CLICK	
Componente	**Predisposizione e/o modifica da effettuare**
Ponticello J2A	Chiudere su GND.
Ponticello J3A	Lasciare aperto (floating).
Ponticello J4A	Chiudere su GND.
Ponticello PWR SEL	Collegare in modo che l'alimentazione giunga al pin 7 dello zoccolo del modulo (3,3 V).
Resistore R1 (2,2 Kohm)	Asportare in modo che il LED non aggiunga un inutile consumo di corrente.

Tabella 3 A

Corrispondenza fra i pin dello zoccolo EEPROM CLICK ed il circuito	
Zoccolo 16 pin EEPROM CLICK	**Collegamento allo schema elettrico di figura 7**
Pin 7 (3,3V)	VBATT (alimentazione positiva)
Pin 8 oppure pin 9 (GND)	Massa (pin 1 di J2)
Pin 12 (SCL)	Pin 14 di U2 (SCL- RC3)
Pin 11 (SDA)	Pin 15 di U2 (SDA- RC4)
Pin 16 (PWM - WP)	Pin 25 di U2 (RB4)

Tabella 3 B

tabella 3 A - B: Modifiche e collegamenti sul modulo EEPROM CLICK.

tratteggiato. Questa particolarità grafica indica che il componente è opzionale. Nel caso l'utente decida di installarla può usare il modulo EEPROM CLICK, prodotto da MIKROELEKTRONIKA, in vendita da I. E. SHOP (sul sito è visibile anche lo schema elettrico). L'uso di questa piccola scheda, equipaggiata con una EEPROM tipo 28C08L e pochi atri componenti, consentirà una semplificazione del montaggio, tuttavia occorre fare attenzione alla piedinatura ed alle predisposizioni "a saldare". La disposizione dei vari ponticelli e le piccolissime modifiche necessarie sono riassunte nella tabella 3A, nella tabella 3B è invece riportata la piedinatura dello zoccolo rispetto ai collegamenti sullo schema elettrico. Il valore delle resistenze di pool-up, necessarie al bus I2C, e del condensatore di shunt presenti sul modulo EEPROM CLICK sono invece simili a quelli presenti sullo schema di figura 7. E' anche possibile l'acquisto del singolo componente, comunque rintracciabile sul sito di I. E. SHOP. Nel caso il lettore voglia provvedere con una EEPROM in suo possesso, ricordi che alcuni componenti non hanno l'alimentazione che può scendere sotto i 5 VDC. Essendo il nostro circuito alimentato con 4 batterie al NiCd da 1,2 V cadauna (totale 4,8 V), è necessaria una EEPROM avente tensione di alimentazione minima almeno pari a 4 V (le batterie purtroppo si scaricano…). Consultare quindi il data-sheet del componente ricordando che, per molte case produttrici, la presenza della lettera "L" nella sigla garantisce un funzionamento con un range inferiore pari a 3,5 V. Per quanto riguarda il funzionamento del bus seriale I2C non vorrei soffermarmi troppo, rimando il lettore che ne vuol sapere di più all'ottimo articolo di M. Paolizzi e S. Silvestri, su F.E: un vero tutorial chia-

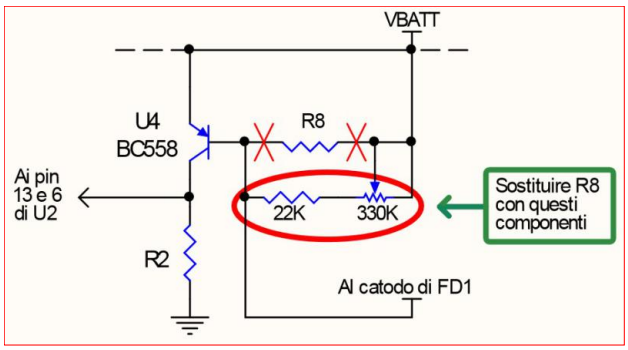


Figura 9: Modifica circuitale per sostituire il fotodiodo BPPW34.

ro e completo (in alternativa consultare le citazioni nel paragrafo "Bibliografia"). Nel nostro progetto, per una maggiore integrità dei dati, è gestita anche la protezione della scrittura sul pin WP. Mentre il riconoscimento da parte del firmware della presenza della EEPROM, come accennato, avviene in maniera automatica: nessuna predisposizione è richiesta. Questa funzione è molto semplice: facendo riferimento allo schema elettrico di figura 7, si noterà la presenza di un resistore da 100 Kohm fra la linea SCL (clock seriale I2C) e la massa. Questo resistore garantisce un livello basso sul relativo pin della CPU se il modulo di memoria o le resistenze di pull-up non sono presenti. Ovviamente in questo caso non è installata alcuna memoria seriale. Se invece il modulo o la memoria sono presenti, e con quest'ultima anche i suoi resistori di pool-up, la linea SCL, per effetto della partizione resistiva (il resistore di pool-up ha un valore molto basso rispetto al resistore da 100 Kohm) sarà a livello alto. E' quindi sufficiente interrogare la linea RC3, prima che essa sia dedicata al bus I2C, per sapere se la EEPROM è presente o meno. Per quanto riguarda il RESET nulla o poco da dire: un semplice circuito RC garantisce la massima affidabilità. Tutti i pool-up delle predisposizioni e dell'unico pulsante sono a livello alto in modo da garantire un assorbimento minimo. Il pulsante, in particolare, è dotato di un semplice circuito RC antirimbalzo (R9 e C14). Il convertitore TTL – RS232 è il "popolare" MAX 232 che, nell'applicazione, è sfruttato in una sola direzione e con una sola porta. Infatti, i dati possono fluire solo dal nostro circuito al PC e non viceversa. Un accenno al fotodiodo infra-rosso, si tratta di un comune ed economico BPW34. Per evitare incertezze sul livello d'ingresso alla CPU, esso è interfacciato da un comune BJT. Questi è stato scelto di tipo PNP in modo che almeno un pin del fotodiodo (l'anodo) possa essere in comune con la linea di massa. Tale accorgimento permette una certa riduzione del rumore che, inevitabilmente e per ragioni diverse (lampade a filamento, segnali R.F., ecc.) troviamo sovrapposto agli impulsi provenienti dal contatore. In ogni caso, il collegamento fra scheda e sonda ottica dovrà essere fatto con cavetto schermato (schermo posto a massa soltanto all'estremità di J2) e non dovrà essere più lungo di circa 30 cm. Non è questa però la sola accortezza. Purtroppo il fotodiodo è sensibile anche a parte dello spettro di luce visibile e la luce stessa, pure se proveniente da sorgenti artificiali, è composta anche da una forte componente di raggi infrarossi, Tutto

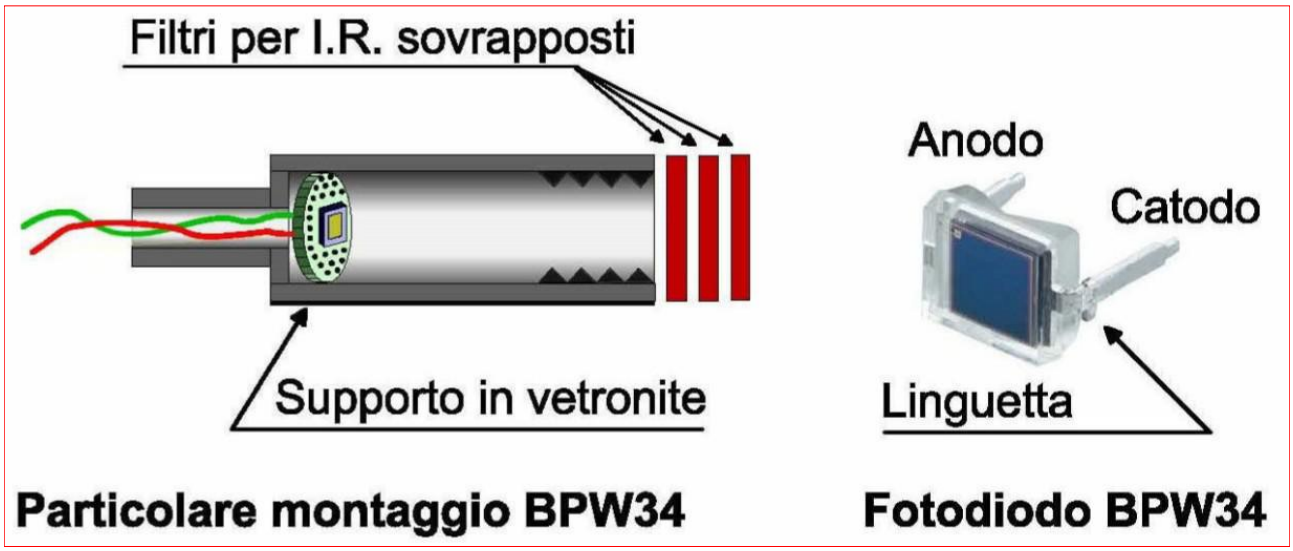


Figura 10: Particolari costruttivi della sonda ottica I. R.

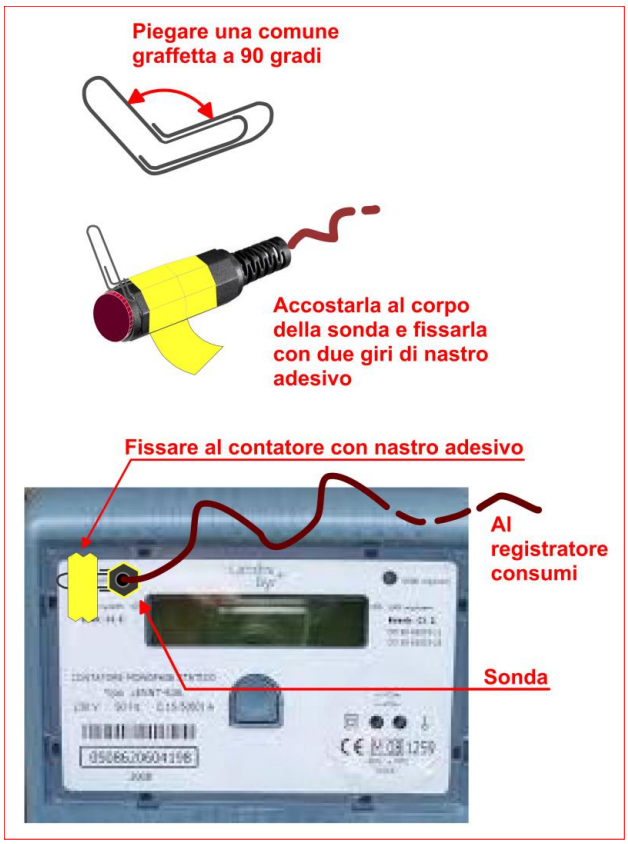


Figura 11:Fissaggio della sonda ottica I. R. al contatore.

questo impone la necessità di un piccolo filtro ottico, viceversa si corre il rischio di saturare, con luce ambiente accidentale, la nostra sonda I. R. impedendogli, di fatto, la corretta rilevazione degli impulsi I. R.. Nel caso il lettore avesse a disposizione altri fotodiodi e volesse sostituirli al BPW34 consiglio una piccola modifica al front-end ottico. Essa è visibile in figura 9. Si tratta in pratica di sostituire il resistore R8 (68 Kohm) con un gruppo resistore e trimmer di valore come in figura 9. La taratura andrà fatta schermando completamente dalla luce il fotodiodo e portando, inizialmente, il cursore del trimmer verso il positivo VBATT. Ponendo un comune tester fra il collettore di U4 e la massa dovremmo poter leggere, in queste condizioni, un livello pari a circa VBATT. In altri termini U4 sta conducendo anche con la debolissima corrente inversa prodotta dal fotodiodo al "buio". Ora, si giri lentamente il cursore del trimmer fino a leggere 0 Volt sul tester. Una volta raggiunta questa condizione, si giri ancora il cursore del trimmer di pochissimi gradi nella stessa direzione. Ciò ci consentirà di stare tranquilli nei confronti di eventuali aumenti di temperatura che, com'è noto, provocano aumenti della corrente inversa. Colleghiamo il tutto al contatore, stabilendo una connessione ottica, e portiamo il nostro dispositivo in modalità TEST MONITOR, Per farlo è sufficiente tenere premuto il pulsante P1 al momento dell'accensione. Se abbiamo un utilizzatore di energia attivo (meglio se di potenza) e tutto è andato bene, dovremmo poter vedere gli impulsi I. R. del contatore ripetuti sul LED rosso LD1. Se ciò non avvenisse possono essere due i casi:

- La taratura va ritoccata spostando leggermente il cursore del trimmer verso la massa.
- Il fotodiodo non è di tipo compatibile.

Accennavamo, poco sopra, alla necessità di un filtro ottico. Io l'ho ottenuto ritagliando la plastica di un vecchio display a LED a sette segmenti di luminescenza rossa. Il tutto è stato poi racchiuso in un contenitore plastico per jack audio miniatura. I piccoli filtri, meglio se 2 o 3 uniti fra loro, sono posti all'imboccatura del jack, Se sono ritagliati di dimensioni adeguate possono benissimo incastrarsi fra le righe della filettatura (vedi figura 10). Il fissaggio al contatore potrà avvenire con l'ausilio di una graffetta, opportunamente piegata, e qualche giro di scotch. La grafica di figura 11 illustra i vari passaggi.

Conclusione

Una volta terminata la costruzione della scheda e della sonda sarà possibile il test dell'insieme con le modalità viste al paragrafo precedente (TEST MONITOR). Ricordarsi comunque di impostare il DIP_FIX JP1 secondo quanto riassunto nella figura 4 (vedi) e di avviare qualche elettrodomestico per avere un certo consumo di energia. Nella prossima (ed ultima) puntata affronteremo le caratteristiche del firmware, l'interfaccia utente e la raccolta dei dati su foglio EXCEL.

Bibliografia

"I2C-bus specification and user manual" UM10204 Rev. 6 (NXP Semiconductors).

GESTIONE DI UNA CISTERNA PER LA RACCOLTA ACQUA PIOVANA CON ARDUINO

di Girolamo D'Orio

Trattiamo qui della realizzazione di un dispositivo per la gestione di una cisterna per acqua piovana. La visualizzazione del livello avviene su display LCD 2x16. Tramite rotary encoder è possibile effettuare la scelta del quantitativo riempibile in caso di siccità con elettrovalvola per afflusso di acqua diretta. I dati impostati sono salvati su Eeprom. Si può escludere la pompa aspirante quando il livello dell'acqua è minimo.

Questa realizzazione è nata da una mia esigenza che ho dovuto affrontare nell'estate siccitosa del 2015. Purtroppo i mutamenti climatici hanno la tendenza a manifestarsi con caratteristiche di un clima tropicale, nel nostro paese. Lunghi periodi di siccità vengono interrotti da violenti temporali e nubifragi.

Nel mio lavoro, nel periodo da Aprile ad Agosto, occorre avere sempre una notevole disponibilità di acqua per eseguire i trattamenti nel vigneto. Anche con le nuove attrezzature, come gli atomizzatori a basso volume, occorrono almeno 200 litri di acqua per ettaro di vigneto per effettuare un trattamento per ottenere una buona copertura. Purtroppo l'acqua potabile ha un costo e l'obbiettivo è di consumarla il meno possibile per il suo costo e il relativo rispetto civico.

L'installazione di una cisterna per l'accumulo di acqua piovana raccolta da un tetto è un'ottima soluzione, ma se non piove come la riempiamo? Per anni ho fatto come, penso, fanno un po' tutti in caso di siccità: la classica

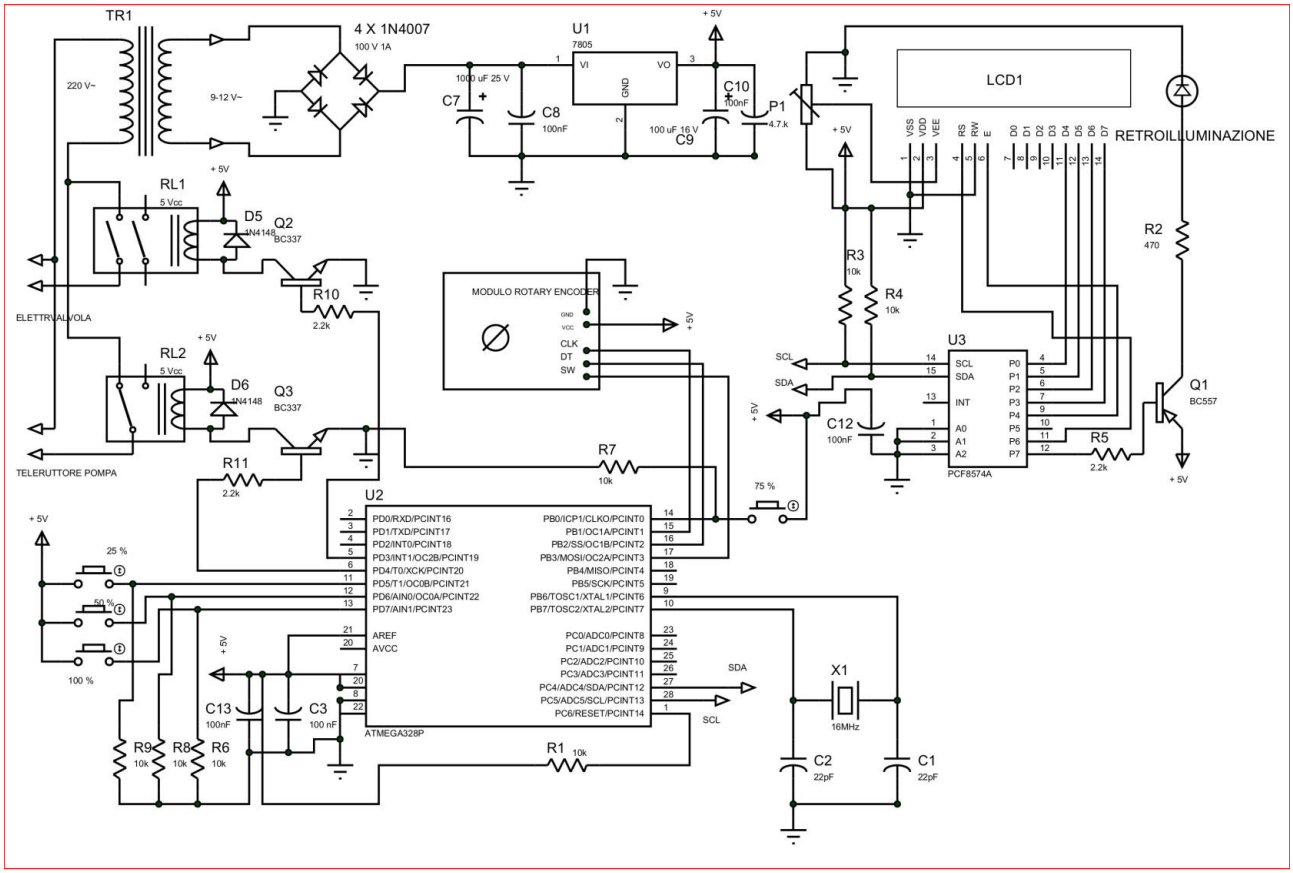


Figura 1: Schema elettrico.

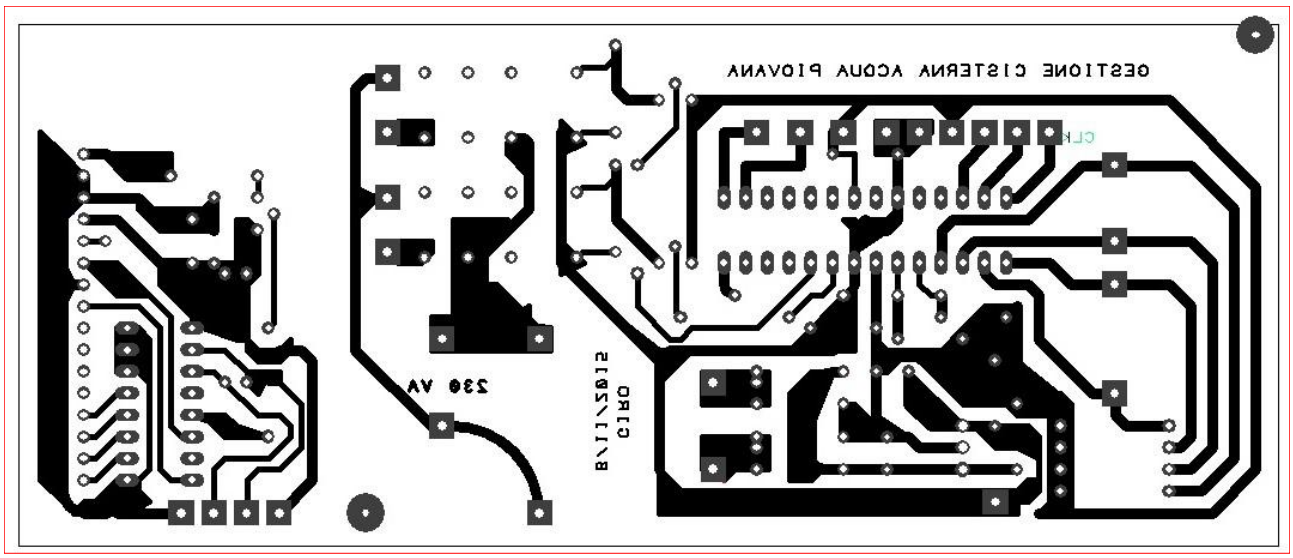


Figura 2: PCB (148 mm x 61 mm).

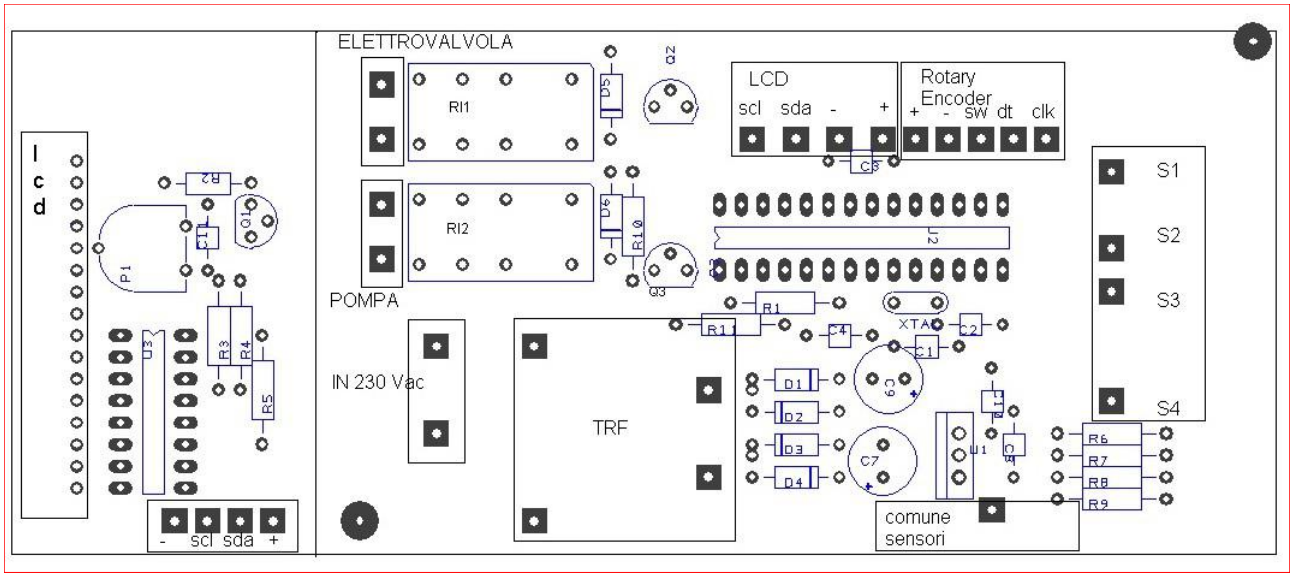


Figura 3: Disposizione componenti.

tubazione in gomma “volante” per riempire la cisterna. Spesso la portata è minima, occorre molto tempo per arrivare al livello desiderato per fronteggiare l’emergenza. Il problema è quindi chiuderla al momento giusto per evitare sprechi di acqua. Il dispositivo creato serve proprio a questo.

Quattro interruttori galleggianti, opportunamente installati, rilevano il livello con uno step del 25% rispetto alla capacità massima. Gli interruttori galleggianti non solo comunicano costantemente il livello della cisterna su lcd, ma svolgono la funzione anche di interruttore di minimo e di massimo per la fase riempimento. Il valore di riempimento massimo è configurabile. Tramite rotary encoder, tenendolo premuto almeno per circa 1 secondo, entreremo nel menù per scegliere fino a quale percentuale la cisterna dovrà essere riempita con acqua potabile. Il dato scelto verrà immagazzina nella eeprom del microcontrollore, così il dispositivo, anche in caso di momentaneo blackout svolgerà correttamente il suo compito. Ad ogni riavvio, il dispositivo, visualizza su lcd, durante la fase di inizializzazione, il livello di riempimento precedentemente impostato, per evitare tutte le volte di selezionarlo e di continuare a fare il suo lavoro anche dopo il blackout momentaneo. La mia

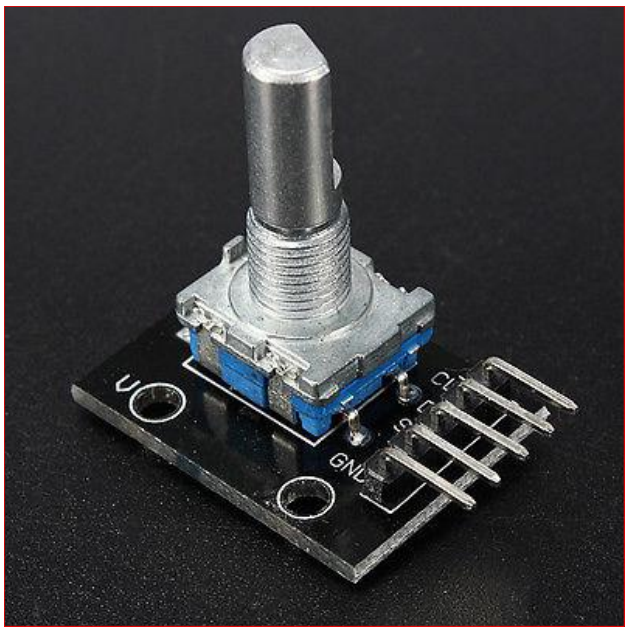


Figura 4: Encoder rotativo.

esigenza è di avere sempre una cisterna che abbia un livello minimo da permettermi di svolgere un trattamento, non mi devo trovare nella condizione che mi manchi acqua. Ognuno di noi avrà la propria esigenza, quindi il sorgente dovrà essere compilato inserendo dati corretti per ottimizzare le vostre necessità.
Ecco la mia condizione per farvi capire come dovrà essere impostato, per arrivare all'obbiettivo di sprecare inutilmente acqua potabile.
La cisterna ha una capacità di 150 hl e ad ogni trattamento mi occorrono circa

Figura 5: Interruttore galleggiante di livello liquidi.

Elenco componenti

R2	470 Ω ¼ W
R1,R3, R4,R6-R9	10 KΩ ¼ W
R5,R10,R11	2,2 KΩ ¼ W
C7	1000uF 24V elettr.
C9	100uF 16V elettr.
C1,C2	22 pF poliestere
C3,C4,C8, C10,C11	100 nF poliestere
U1	7805
U2	ATmega328p-pu
U3	PCF8574N
D1-D4	1N4007
D5,D6	1N4148
Q1	BC557 PNP
Q2,Q3	BC337NPN
XTAL	16 Mhz
TR1	Trasformatore 0-9v 700mA

80 hl. Il trattamento necessità di 2 giorni lavorativi per effettuarlo. Mettiamoci nella condizione che non è prevista pioggia nei giorni antecedenti al trattamento. Accendo il dispositivo e scelgo che la cisterna dovrà riempirsi con acqua potabile fino al 50% della sua capacità.
L'elettrovalvola a questo punto si apre se la cisterna è al di sotto della soglia imposta, si chiuderà solamente quando l'acqua ha raggiunto il livello del 50%. Il dispositivo rimane quindi acceso e io incomincio a prelevare l'acqua per il mio scopo. L'elettrovalvola rimane chiusa, si apre solamente quando scenderò sotto il livello del 25%. Arrivati a questa soglia si commuta anche il relè che disabilita la pompa di prelievo acqua. Perché questa scelta?
Oltre a salvaguardare la pompa che non lavori in assenza di acqua, se la cisterna, come nel mio caso, è interrata, se rimane vuota, si può andare incontro ad un notevole problema.
Nel caso di forti temporali può capitare che le gronde del tetto non riescano a

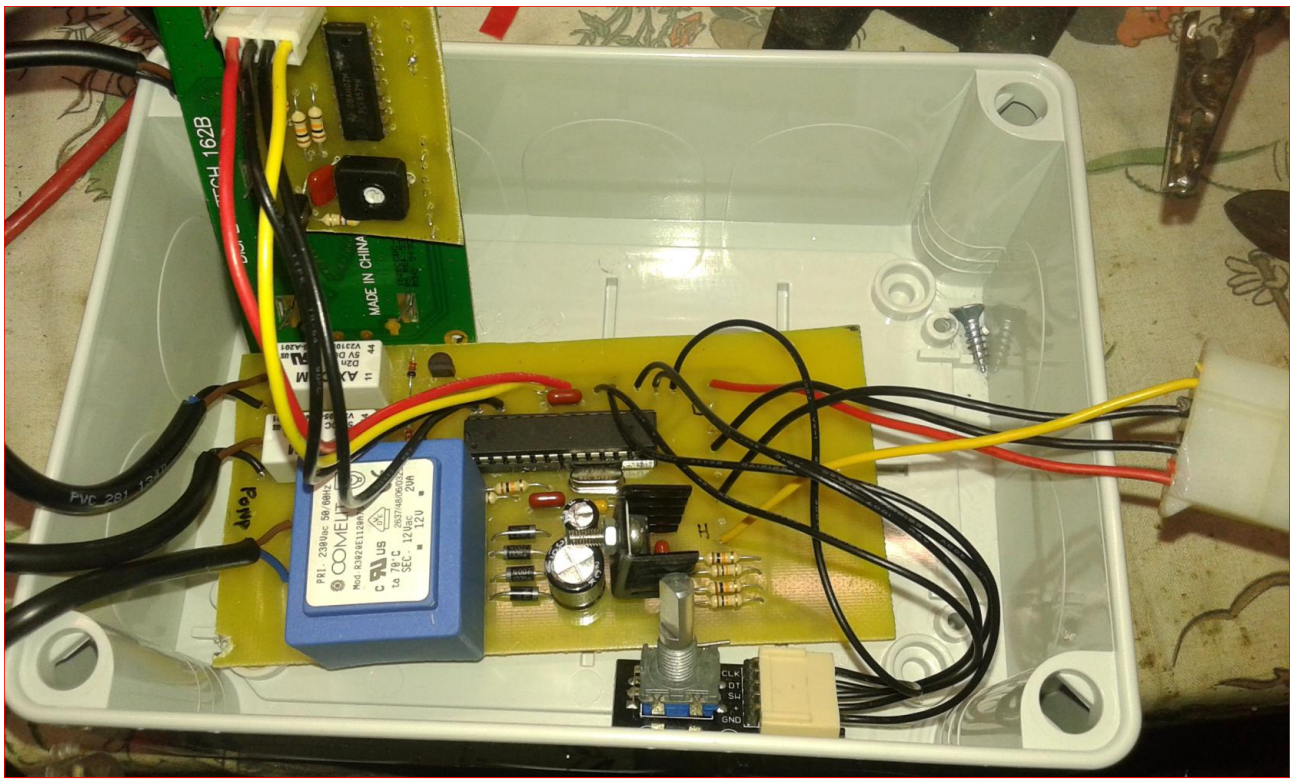


Figura 6: Realizzazione.

far defluire correttamente l'acqua. Molto spesso, quando accadono questi fenomeni, la prima cosa che scorre nei canali, sono i nidi di volatili, foglie o altro che vanno ad ostruire il canale. Quindi, in questo caso, ci troviamo in una condizione dove l'acqua piovana non arriva alla cisterna, ma sta piovendo molto.
Se la cisterna è vuota, e in caso di temporale si infiltra l'acqua dal sottosuolo alla base della cisterna, si verifica il principio fisico che permette alle grandi navi di galleggiare.
In questo caso vi vedrete riemergere dal terreno la cisterna, che sia di cemento, in pvc o di acciaio. Un vero disastro. E' per questo motivo che ho deciso che la cisterna deve almeno avere un livello pari ad ¼ della sua capacità totale.

Descrizione del circuito

Prima di trattare della descrizione hardware, faccio notare che il PCB, in molti punti, ha la presenza di tensione di 230 Vac, quindi occorre prestare molta attenzione, soprattutto in fase di collaudo.
Fate attenzione a dove lo appoggiate, a farsi male sul serio ci vuole poco.
L'alimentatore, costituito dal trasformatore, dal ponte a diodi, dallo stabilizzatore a 5Vcc e contornato dalle capacità di livellamento e di filtraggio, alimenta tutto il circuito.
L'assorbimento, non modestissimo quando capita che anche i due relè sono eccitati, può arrivare ad assorbire oltre 500 mA, quindi si deve "alettare" generosamente il 7805, soprattutto se reperite un trasformatore con un secondario superiore a 9Vac.
Per gli interruttore galleggianti, ho scelto questa soluzione di proposito, in modo da non portare tensioni e correnti a contatto diretto con l'acqua. Lavorando con tensioni continue e non alternate, per non complicare il progetto, il fenomeno dell'elettrolisi si farebbe notare subito con un conseguente errore di lettura del livello.
Questi galleggianti-interrutori al loro interno hanno un reed e aggirano que-

Figura 7: installazione dei galleggianti-interruttori su un profilato.

sto problema, e dato il loro limitato costo (si parla di pochi euro) ho pensato che sia un giusto compromesso costo-beneficio. Gli ingressi al microcontrollore ATmega328 li ho configurati in come se fossero comuni pulsanti, quindi, per come ho scritto nel sorgente, li ho equipaggiati con resistenze di pulldown da 10K. L'encoder rotativo, modulo già assemblato, può essere collegato direttamente al microcontrollore.

Per LCD 2x16, la comunicazione avviene tramite I2C, passando dall'integrato I/O expander PCF8574n. Con questa comunicazione semplifico cablaggi dal micro a lcd e risparmio preziose porte di I/O.

Il piccolo circuito separato può essere saldato direttamente dietro l'LCD, in modo di avere solo quattro collega-

Figura 8: Vista dall'alto dell'interno della cisterna inox interrata.

menti: alimentazione +/- e comunicazione SDA/SCL. Come avete notato dallo schema, gli scambi dei relè sono NA e NC.

Ho fatto questa scelta per un risparmio di consumo di energia elettrica. Si presuppone che la cisterna sia almeno sempre sopra il livello minimo, quindi il relè che comanda la pompa RL2 è usato in NC, mentre l'attivazione dell'elettrovalvola si presuppone che sia un uso limitato, quindi il contatto è in NA. Tutto ciò per tenere il meno possibile le bobine dei rispettivi relè eccitate.

Descrizione del software per Arduino

Il sorgente è ben commentato e non troverete nessuna difficoltà nel capirlo. Per prima cosa richiamo le librerie installate per salvare i dati in eeprom e per il dialogo in I2C per LCD. In LCD compare costantemente il livello di acqua presente nella cisterna.

Premendo il pulsante sul rotary encoder entreremo nel menù al fine di scegliere il livello a cui la cisterna dovrà

essere riempita con acqua potabile.
Ruotando il rotary encoder su LCD compare la percentuale di riempimento, che andrà confermata premendo nuovamente il pulsante.
Fatto ciò, il dato viene salvato in eeprom, al prossimo riavvio il software andrà a recuperarlo, quindi niente paura per eventuali blackout.
Quando l'elettrovalvola si apre, perché il software la abilita in base ai parametri fissati, nella seconda riga di LCD compare la voce "RIEMPIMENTO". Ricordo ancora una volta che, una volta raggiunto l'obbiettivo di riempimento impostato precedentemente, l'elettrovalvola che permette il riempimento non si apre fino a quando l'interruttore-galleggiante di minimo non si apre. Quando si apre questo ultimo interruttore commuta anche il relè che disabiliterà la pompa di aspirazione dell'acqua della cisterna.
Quindi per vostre esigenze diverse, lavorate sul sorgente o sul posizionamento dei galleggianti-interruttori.

Consigli per il montaggio

Il montaggio non presenta particolari difficoltà. Per i collegamenti con l'esterno ho utilizzato dei connettori di recupero in disuso. Per collegare i galleggianti-interruttori ho usato un cavo multipolare tipo un fror 5x1.5mm.
La sezione non è modesta, considerando che la corrente che circola è prossima allo zero.
La scelta è dovuta al fatto di limitare la caduta di tensione, considerando che la tensione è 5Vcc e nel mio caso la lunghezza di linea tra PCB e i galleggianti-interruttori si aggira intorno ai 15 Mt.
Per l'elettrovalvola a 230 Vac da almeno ½ pollice, il collegamento va effettuato con un 3x1.5 e si deve collegare il connettore di messa a terra predisposto sul essa.
Consiglio di usare l'uscita del relè che

Figura 9: Foto del dispositivo installato.

abilita/disabilita la pompa, per comandare direttamente la bobina di un teleruttore, il quale accende/spegne la pompa stessa, se quest'ultima dovesse avere un generoso assorbimento.

Conclusioni

Questo semplice dispositivo, con semplici correzioni al sorgente, può essere impiegato in notevoli applicazioni, non solo ad uso agricolo-professionale, basta pensare all'irrigazione del giardino di casa. Cerchiamo di ottimizzare con efficienza una risorsa importante come l'acqua. Purtroppo il suo maggiore utilizzo avviene quando non piove. Quindi, con un occhio alle previsioni meteo e all'utilizzo intelligente di questo piccolo dispositivo, facciamo sicuramente del bene al portafoglio e all'ambiente. Buon divertimento e buona realizzazione a tutti.
Giro.

LISTATO

```
#include <EEPROM.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x20,16,2);  // 0x38 utilizzando il pcf8574AN con A1,A2,A3 a gnd---
-0x20 utilizzando il pcf8574N con A1,A2,A3 a gnd

 // PIN 10 SW PULSANTE NORMALMENTEAPERTO LIVELLO LOGICO ALTO
 int val;
  int encoder0PinA = 9;// CLK
  int encoder0PinB = 10;// DT
  int encoder0Pos = -1;
  int encoder0PinALast = LOW;
  int n = LOW;
  int menu; //variabile di comodo per entrare nel menu scelta riempimento
  int livello;
  int range;
void encoder(){ //menu per selezionare il livello di riempimento
   n = digitalRead(encoder0PinA);
    if ((encoder0PinALast == LOW) && (n == HIGH)) {
      if (digitalRead(encoder0PinB) == LOW) {
        encoder0Pos--;
      } else {
        encoder0Pos++;
      }
      Serial.println (encoder0Pos);
    if ((encoder0Pos<-4) || (encoder0Pos>4)){
      encoder0Pos=0;

    }
    }
       if (encoder0Pos <0){
        encoder0Pos=(encoder0Pos * -1);

      }
       livello=encoder0Pos;

lcd.setCursor(0,1);

if (livello==0){
            lcd.print("  0%");
}
if (livello==1){
            lcd.print(" 25%");
}
if (livello==2){
            lcd.print(" 50%");
}
if (livello==3){
            lcd.print(" 75%");
}
if (livello==4){
```

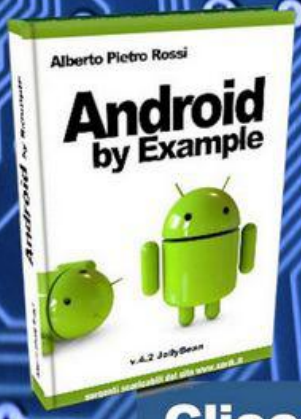
```
            lcd.print("100%");
}


 //      lcd.setCursor(0,1);
 //   lcd.print(encoder0Pos);
    encoder0PinALast = n;
     if (digitalRead(11)==LOW){
    delay(500);
    EEPROM.write(1, livello); // memorizzo nella eeprom la percentuale di livello scelta
delay(50); // piccola pausa per assicurare la corretta scrittura
    menu=0;
  }
}
void setup()
{
 pinMode(5,INPUT);//25%
 pinMode(6,INPUT);//50%
 pinMode(7,INPUT);//75%
 pinMode(8,INPUT);//100%
 pinMode (encoder0PinA,INPUT);//9
    pinMode (encoder0PinB,INPUT);//10
 pinMode(11,INPUT);//PULSANTE ENCODER
  pinMode(4,OUTPUT);// RELè USANDO CONTATTO NC PER DISATTIVARE POMPA ASPIRAZIONE
 pinMode(3,OUTPUT);// ELETTROVALVOLA apertura acqua

 digitalWrite(13,LOW);

   Serial.begin (9600);
  lcd.init();                      // inizializzazione  lcd


  lcd.backlight();
   lcd.setCursor(0,0);
  lcd.print("LIVELLO CISTERNA");
    lcd.setCursor(0,1);
  lcd.print("     HL 150     ");
  delay(2000);

  delay(300);

  lcd.setCursor(5,1);
  lcd.print("BY GIRO");

  delay(2000);
   lcd.clear();
    livello=EEPROM.read(1);// lettura dalla eeprom del livello precedentemente impostato
  delay(50);
   lcd.setCursor(0,0);
  lcd.print("LIV. RIEMPIMENTO");
   lcd.setCursor(5,1);
 if (livello==0){
            lcd.print("  0%");
}
```

```
if (livello==1){
            lcd.print(" 25%");
}
if (livello==2){
            lcd.print(" 50%");
}
if (livello==3){
            lcd.print(" 75%");
}
if (livello==4){
            lcd.print("100%");
}
delay(2000);
lcd.clear();
}

void loop() {
  if (digitalRead(11)==LOW){// pulsante encoder per entrare nel menu
    delay(500);
    menu=1;
     lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("  % RIEMPIMENTO");
  }
  while (menu==1){

    encoder();
    if (menu==0){
     lcd.clear();
      break;
    }
  }
 if ((digitalRead(5)==LOW)&&(digitalRead(6)==LOW) && (digitalRead(7)==LOW) &&
(digitalRead(8)==LOW)){
  digitalWrite(4,HIGH); // DISABILITO LA POMPA ASPIRANTE
lcd.setCursor(0,0);
lcd.print("LIVEL:   <35  Hl");
if (livello==0){
 digitalWrite(3,LOW);
}
   else{
    lcd.setCursor(0,1);
lcd.print("  RIEMPIMENTO...");
digitalWrite(3,HIGH);
   }
  }
  if ((digitalRead(5)==HIGH)&&(digitalRead(6)==LOW) && (digitalRead(7)==LOW) &&
(digitalRead(8)==LOW)){  // siamo al 25%
    digitalWrite(4,LOW); //ABILITO POMPA ASPIRANTE
    lcd.setCursor(0,0);
lcd.print("LIVEL: 35-75  Hl");
if ((digitalRead(3)==HIGH)&&(livello>=2)){
      lcd.setCursor(0,1);
lcd.print("  RIEMPIMENTO...");
```

```
}
else {
    lcd.setCursor(0,1);
lcd.print("                ");
digitalWrite(3,LOW);
}
  }

  if ((digitalRead(5)==HIGH)&&(digitalRead(6)==HIGH) && (digitalRead(7)==LOW) &&
(digitalRead(8)==LOW)){  // siamo al 50%
  digitalWrite(4,LOW);// ABILITO POMPA ASPIRANTE
    lcd.setCursor(0,0);
lcd.print("LIVEL: 75-110  Hl");
if ((digitalRead(3)==HIGH)&&(livello>=3)){
      lcd.setCursor(0,1);
lcd.print("  RIEMPIMENTO...");
}
else {
    lcd.setCursor(0,1);
lcd.print("                ");
digitalWrite(3,LOW);
}
  }

  if ((digitalRead(5)==HIGH)&&(digitalRead(6)==HIGH) && (digitalRead(7)==HIGH) &&
(digitalRead(8)==LOW)){  // siamo al 75%
  digitalWrite(4,LOW);// ABILITO POMPA ASPIRANTE
    lcd.setCursor(0,0);
lcd.print("LIVEL:110-150  Hl");
if ((digitalRead(3)==HIGH)&&(livello>=4)){
      lcd.setCursor(0,1);
lcd.print("  RIEMPIMENTO...");
}
else {
    lcd.setCursor(0,1);
lcd.print("                ");
digitalWrite(3,LOW);
}
  }

    if ((digitalRead(5)==HIGH)&&(digitalRead(6)==HIGH) && (digitalRead(7)==HIGH) &&
(digitalRead(8)==HIGH)){  // siamo al 100%
  digitalWrite(4,LOW); //ABILITO POMPA ASPIRANTE
    lcd.setCursor(0,0);
lcd.print("LIVEL:   150  Hl");

    lcd.setCursor(0,1);
lcd.print("                ");
digitalWrite(3,LOW);
}

 delay(1000);

  }//end loop
```

SEGUICI SU

facebook



Clicca "Mi Piace"
e scarica subito la
Guida Android

FARE ELETTRONICA

DIRETTORE RESPONSABILE
Antonio Cirella

DIRETTORE TECNICO
Giovanni Di Maria

Hanno collaborato in questo numero:
Ing. Giovanni Carrera, La Rosa Giuseppe, Lucio Di Jasio, Andreas Mangler, Grazia Ancona, Girolamo D'Orio.

Direzione Redazione:
CONSORZIO TECNOIMPRESE SCARL
Via Console Flaminio, 19
20134 MILANO – MI

FARE ELETTRONICA

© Copyright
Tutti i diritti di riproduzione o di traduzione degli articoli pubblicati sono riservati.
Manoscritti, disegni e fotografie sono di proprietà di **CONSORZIO TECNOIMPRESE SCARL**. È vietata la riproduzione anche parziale degli articoli salvo espressa autorizzazione scritta dell'editore. I contenuti pubblicitari sono riportati senza responsabilità, a puro titolo informativo.